A COMPARISON OF TWO ADAPTIVE PREDICTION SYSTEMS

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

By

Yet - Ghee Yap

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in the School of

Industrial and Systems Engineering

Georgia Institute of Technology

June, 1973

A COMPARISON OF TWO ADAPTIVE PREDICTION SYSTEMS

Approved:

Dr. Douglas C. Montgomery, Chairman

Dr. Richard H. Deane

Dr. Russell G. Heikes

Date approved by Chairman: June 5, 1973

ACKNOWLEDGMENTS

The author wishes to express her sincere thanks for the valuable criticism, guidance, and encouragement given by Dr. Douglas C. Montgomery who acted as thesis advisor.

Appreciation is extended to the thesis committee, Dr. Richard H. Deane and Dr. Russell G. Heikes for their constructive comments and support during this study.

A special "thanks" is extended to Mr. John Smith for helping to develop the programs for the calculator plotter.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

SUMMARY

The technique of exponential smoothing has received much attention in the recent years. It is one of the most popular naive forecasting schemes, but unfortunately the smoothing constants are selected quantitatively through some crucial assumptions about the time series. To compensate for this defect, two evolutionary operation (EVOP) techniques have been developed to make the exponential smoothing parameters self-adaptive.

The two evolutionary operation procedures are the standard factorial design EVOP procedure, and the sequential application of the simplex design. They will be used to make the Winter's model self-adaptive. The model is made self-adaptive in the sense that the parameters in the model are automatically adjusted to compensate for changes in the basic nature of the time series. A direct comparison is made between the two methods through simulation.

Eight time series, possessing certain characteristics and with different magnitudes of the random components superimposed on them, are generated. The factorial EVOP and the simplex EVOP are then used to predict the eight time series. The general conclusions obtained are as follows:

1. Simplex EVOP is more sensitive to the choice of the initial values of the smoothing constants.

2. Factorial EVOP performs better than the simplex EVOP when

there is a presence of some periodic factor in the series.

3. Simplex EVOP performs better than the factorial EVOP when there is a presence of a trend factor in the series.

CHAPTER I

INTRODUCTION

Forecasting the value of a time series using available observations can provide an important basis for (a) economic and business planning, (b) production planning, (c) inventory and production control and (d) control and optimization of industrial processes. Therefore, the problem of making good forecasts is of special interest to managers, salesmen, economists, engineers and others who are concerned with quantity and quality of industrial products. Moreover, almost all economic and business decisions depend on forecasting to reduce the doubts and uncertainties in decision making. Forecasting is then of vital importance to the successful operation of business enterprises.

Methods of short-term forecasting were classified by the late Charles S. Roos (7), one of the founders of the econometric society, into five categories: (1) naive methods, (2) leading indexes, (3) comparative pressures, (4) opinion polls, and (5) econometrics. This thesis will be concerned with scientific methods of forecasting future values of a time series based only on past data. Thus, the methods employed here are classified as naive.

Of the many methods of naive forecasting, exponential smoothing is one of the most well known and most successful. One of the main reasons for its widespread use is that it is accurate, since the model is fitted to the data by least squares, with the criteria of fit being discounted in time. Also, it is efficient computationally, since one does not need to retain considerable historical data from one forecast to another. Finally, it is effective in fitting a wide class of models to a set of data.

Statement of the Problem

Regardless of the exponential smoothing model chosen, the ability of the forecasting system to track changes in the time series depends mainly on the choice of the smoothing constant. When the smoothing constant is small, say close to zero, more weight will be associated to the historical data. If the smoothing constant is large, say close to one, more weight will be placed on the current observations. Since exponential smoothing always requires an initial value of the smoothing constant to start the process, one is forced to choose an initial "optimal" smoothing constant. One of the disadvantages of exponential smoothing is that it suffers from the inability to select the smoothing constants quantitatively without making some rigid assumptions about the time series.

Another problem arises when the system encounters sudden changes in the underlying process. Then it will take an unacceptably long time for the system to adjust to, or track, the new signal. As a result, biased forecasts will occur, and continue for some time. Such a situation can be easily detected by the tracking signals. When the tracking signal goes out of control, one can then manually intervene, review the process and start the smoothing procedure all over again. However, when forecasts are being made regularly for many different time series, it is very difficult to manually intervene effectively. It may also be prohibitively expensive. Consequently, effort has been made directly toward developing a system that will automically monitor the smoothing constant, and change its value when the parameters in the underlying time series model change.

Objective of the Thesis

The objective of this thesis is to investigate two self-adaptive smoothing systems. Two evolutionary operation techniques will be used to make Winter's (10) model self-adaptive. The model is made self-adaptive in the sense that the parameters in the model are automatically adjusted to compensate for changes in the basic nature of the time series. Results obtained from the two self-adaptive systems will be compared. The specific thesis objectives are:

1. To determine which evolutionary operating technique is best. The criterion of optimality is the square of the forecast error.

2. To analyse the rate of response to standard input signals, such as step, impulse and ramp functions.

3. To determine whether the systems are sensitive to the choice of initial values of the smoothing constants and the size of the experimental designs used in the control procedure.

Survey of Adaptive Prediction Systems

Several examples of adaptive predicting or forecasting systems are found in the literature. These systems are predominantly used in forecasting sales or product demand. These systems do not necessarily use Winter's model as the basic forecasting technique, but the adaptive nature of these systems are similar in the sense that they are based on the variation of parameters in the system in response to changes in the nature of the time series of interest. We shall briefly describe four major self-adaptive systems.

Chow (3) has developed a simple exponential model with linear trend correction assumed. He proposed that three forecasts be used, based on three different smoothing constants which are set at high, normal and low levels. To start the process, the smoothing constants are arbitrarily chosen and the actual forecast is made at the normal level. However, when one of the outer values of the smoothing constants yields a better forecast on the basis of an error criteria, the next forecast will be made based on this new "best" value. This new smoothing value is then established as the normal value for the coming time period. High and low smoothing constant values are re-established about the new normal value and the process is repeated.

Trigg and Leach (9) proposed a method of making the system self-adaptive by automatically varying the smoothing constant according to the value of the tracking signal. The tracking signal will fluctuate around zero, if the system is in control. However, if biased errors occur, the value of the tracking signal will move towards plus or minus unity, according to the direction of bias. In order to achieve the self-adaptive response rate, Trigg and Leach set the smoothing constant equal to the modulus of the tracking signal. In this way, the value of the smoothing constant will increase when forecasts go out of control so as to give more weight to the recent observations. Once the system has accommodated the new situation, the value of the smoothing constant is reduced, so as to give weight to past data.

Roberts and Reed (6) have developed a self adaptive forecasting technique (SAFT) which combines the exponential forecasting models of Winter with a response surface analysis technique to test the effects on forecast accuracy of varying the exponential smoothing parameters in the forecasting model. The basic exponential procedure uses a two level factorial design, in which each smoothing constant will be held at a low and a high level. In addition to the factorial points, a center point is added to the design to determine the curvature of the response surface. If an effect of varying one or more of the smoothing parameters is statistically significant, then the center point of the experimental design is shifted accordingly. Successive forecasts are the parameter combination defining the center point of the experimental design. Whenever, the center point is moved, the combination of smoothing parameters used in computing the forecast is changed.

Montgomery (4) has also proposed an evolutionary operation scheme for the adaptive control of exponential smoothing parameters. However, the chosen experimental design is different from that used by Roberts and Reed. He recommends the use of the simplex, which is an orthogonal first order experimental design requiring only one more observation than the number of variables under investigation. The procedure involves changing the exponential smoothing parameters each period by the sequential application of the simplex design. A new simplex is formed each period by deleting only one point from the previous simplex and adding one new point as defined by fixed relationships. The point that is deleted each period is the parameter combination which yields the forecast resulting in the largest forecast error. Thus the design will, theoretically, insure that the forecasting system will traverse the parameter space from points of high forecast error to points of lower forecast error.

Roberts and Reed have shown their work to be superior to those of Chow and Winter. Montgomery has shown his procedure to be better than that of Chow. Since the main objective of this study is to compare the work of Roberts and Reed and that of Montgomery, detailed description of their schemes will be discussed in the next chapter.

CHAPTER II

PSEUDO EVOLUTIONARY OPERATION PROCEDURES

This chapter will present a detailed description of the two response surface technqies which will be investigated in this thesis. The name "response surface" was coined by Box (1) to denote surfaces which are formed by the "response" of a certain criterion from various combinations of environmental or independent factors. In this thesis the square of the forecast error is used as the criterion and the exponential smoothing parameters are the environmental factors. The square of the forecast error was chosen for several reasons. It is most commonly used in forecasting as a criterion to be minimized. For response surface analysis, the square of the error always insures values of the response surface greater than or equal to zero. It also places more emphasis on larger errors which tend to make the convex surface more pronounced. The forecast error is defined as

$$E\ (t + 1) = F\ (t + 1) - X\ (t + 1)$$

where $F(t + 1)$ is the one-period ahead forecast made at time t, and $X(t + 1)$ is the actual observation of the time series at time $t + 1$, Thus, square of the forecast error is then equal to $E\ (t + 1)^2$.

Several response surface methods are used in various kinds of practical problems; however, the methods to be investigated here are known as Evolutionary Operation (EVOP). These techniques imitate the natural evolutionary process described by Box (1) in that they consist of systematically introducing variation in selected independent variables which affect the process, and then in some manner select the best operating conditions. In this way, information is produced in a systematic manner and the results are immediately applied. The perturbations introduced through EVOP are aside from normal process variability; and from information gleaned through this variation, EVOP gradually pushes the process toward its optimal operating conditions.

Something to point out that the philosophy here is not the same as the original EVOP. We have replication of each design point at every period, and original EVOP does not. Also, theoriginal EVOP assumes that only small changes in the independent variables can be made, while we can make any sort of changes in the smoothing constants as we wish. For short, we will call our pseudo-EVOP procedure EVOP.

Two EVOP techniques are to be investigated. One procedure is just the usual EVOP described by Box (1) and more recently by Box and Draper (2), which utilizes the 2^k factorial experimental design. The other technique, described by Spendley (8), utilizes the simplex design.

Factorial EVOP

The forecasting model determines forecasts based on various values of the smoothing constants from the experimental design. As soon as the actual observation is determined, the response surface for the combination of the smoothing constants is formed according to the square of the forecast error. Then, using the modified EVOP, the effects of the smoothing constants are tested at 99 percent confidence levels for significant changes. If the changes are significant, the design is then adjusted. Consequently, the smoothing constants are changed, resulting a redefined forecasting system.

Three cases may be developed from this approach:

1. A one-parameter system, where there is no trend or seasonality.

2. A two-parameter system where there is either a trend or a seasonal factor only.

3. A three-parameter system, which is for both trend and seasonal models.

However, we are only interested in the two-parameter and the three-parameter systems. The actual operations of these two systems will be described in detail.

The two-parameter model is used to forecast time series which can be described by a two-parameter system. The forecasts are made in accordance with the 2^2 factorial design with a center point as shown in Figure 1.

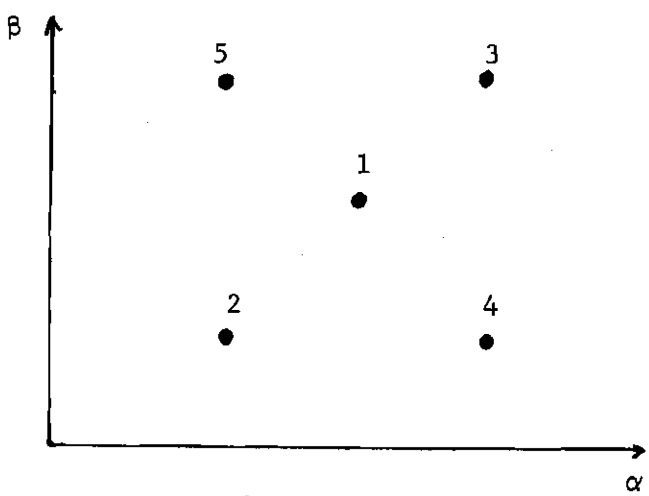


Figure 1. 2^2 Factorial Experimental Design.

The effects are:

Effect of α $= \frac{1}{2} (\bar{r}_3 + \bar{r}_4 - \bar{r}_5 - \bar{r}_2)$

Effect of β $= \frac{1}{2} (\bar{r}_3 + \bar{r}_5 - \bar{r}_2 - \bar{r}_4)$

where $\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_5$ are the average response surface values at the respective experimental design points. It will be noticed here that it is necessary only to determine main effects, since no meaningful information would result from an analysis of the interactions. The 99 percent confidence limits are:

$$\pm 3 \sqrt{\frac{1}{n}} \quad s$$

when n is the cycle number and s is the standard error of the response surface. If either or both effects turn out to be significant the experimental design is shifted in the direction indicated. The upper and lower bounds of the smoothing constants are 0.95 and 0.05, respectively.

The three parameter model can be applied to series which contain both trend and seasonality. Thus, three smoothing constants are involved. A 2^3 factorial plus a center point experimental design is employed. It is an extension of the two-parameter system. The associated design matrix is

$$D = \begin{array}{c} \begin{array}{ccc} \alpha & \beta & \gamma \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

Each row in D represents a point at one of the eight vertices of the cube formed about the center point. A one represents a high level of the variable and a zero represents a low level. Figure 2 represents a graphical display of the center point and the factorial design surrounding it. If the forecasting model is operated at each of these eight points, then the corresponding response surface value can be obtained.

The effects of varying each parameter can now be obtained. If the effect or effects are statistically significant at 99 percent level, then it is necessarily to move the center point in such a way to decrease the response. For example, if the effect of α exceeded the positive limit, then it will hopefully decrease the next period's actual forecast error if the value of α is decreased.

The expressions for the effects of the three-parameters are

$$\text{effect of } \alpha = \tfrac{1}{4}[\bar{r}_2 + \bar{r}_3 + \bar{r}_4 + \bar{r}_5 - \bar{r}_6 - \bar{r}_7 - \bar{r}_8 - \bar{r}_9]$$

$$\text{effect of } \beta = \tfrac{1}{4}[\bar{r}_3 + \bar{r}_4 + \bar{r}_6 + \bar{r}_7 - \bar{r}_2 - \bar{r}_5 - \bar{r}_8 - \bar{r}_9]$$

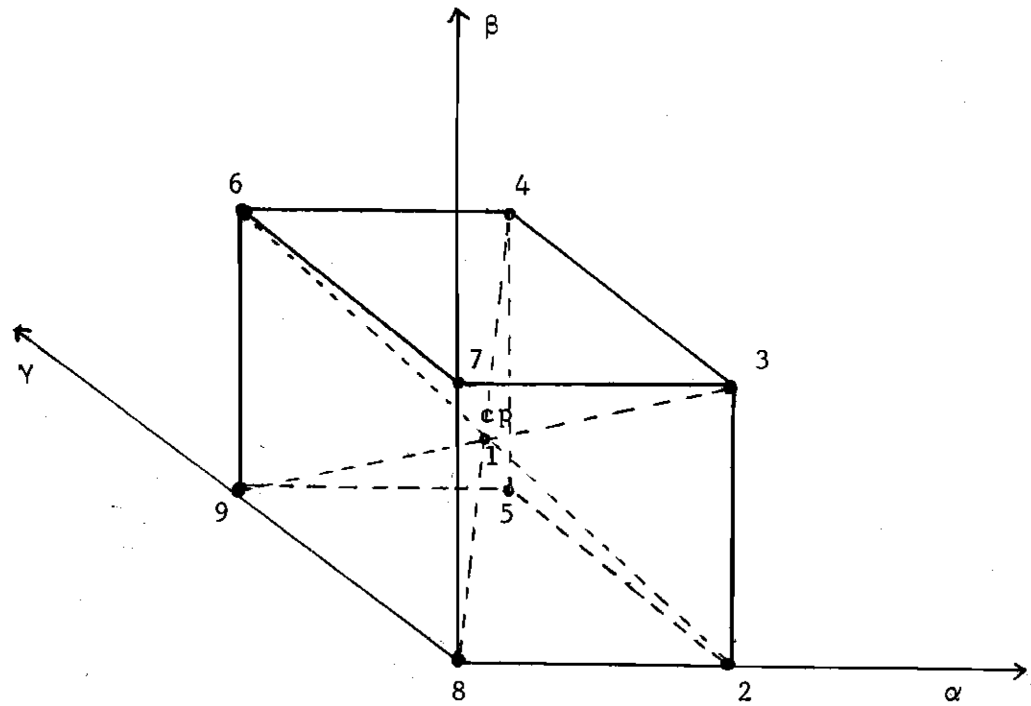
$$\text{effect of } \gamma = \tfrac{1}{4}[\bar{r}_6 + \bar{r}_4 + \bar{r}_5 + \bar{r}_9 - \bar{r}_7 - \bar{r}_3 - \bar{r}_2 - \bar{r}_8]$$

where $\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_9$ are the average response surface values at the respective experimental design points. The limits are

$$\pm 3 \sqrt{\tfrac{1}{2}n} \quad s$$

and the upper and lower bounds for the smoothing constants are 0.95 and 0.05, respectively.

This 2^3 factorial design differs somewhat from the usual three parameter EVOP involves blocking in order to eliminate time effects



cp = center point

Figure 2. 2^3 Factorial Experimental Design.

ordinarily found in industrial applications. EVOP was mainly developed for the chemical industry and since their experimentation involves changing the operating condition of a process over time, the data may be affected by the time effects in the process. This is especially true where considerable time passes between perturbations. However, in the production and control problem, time effects do not exist, since all responses can be determined simultaneously.

Sequential Simplex EVOP

An alternative method of Evolutionary Operation is the sequential simplex technique. This technique was first proposed by Spendley (8) and also by Box and Draper (2). An application of the simplex EVOP technique to sales forecasting is given by Montgomery (4). The sequential simplex technique is basically a much simpler technique than factorial EVOP, both conceptually and computationally.

A simplex is an orthogonal first order experimental design, which requires only one more observation than the number of variables under investigation. Thus, if two smoothing parameters are being controlled the resulting simplex is an equilateral triangle and for a three-parameter smoothing model the simplex is a tetrahedron. A simplex design is an experimental design in which the design points are located at the vertices of a simplex. To illustrate the basic approach of the sequential simplex technique, we shall consider the case of two-parameter smoothing. The simplex, as mentioned above is an equilateral triangle with vertices labelled as 1, 2, 3 as shown in Figure 3.

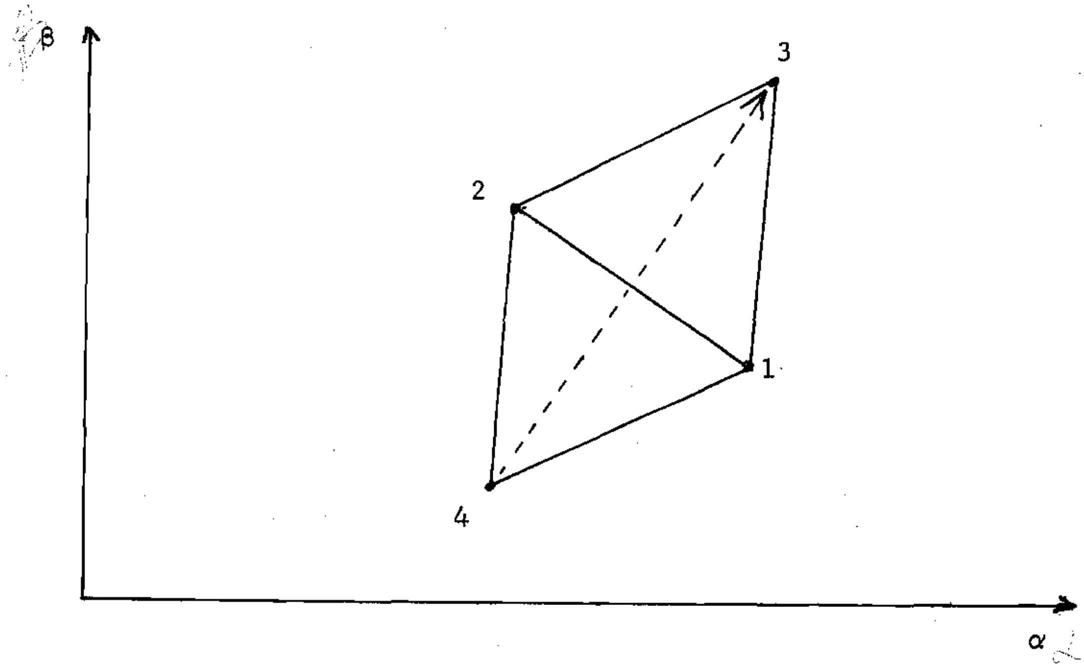


Figure 3. 2 Parameter Simplex Design.

Suppose the value of the forecast error for the three runs, one at each vextex, if found to be greatest at point 3. The simplex procedure will then delete point 3 and add a new point labelled as 4 in Figure 3. This new point is the mirror image of the old point 3. Therefore, point 4 will form an equilateral triangle with the two original points.

The basic design employed is the regular simplex in k dimensions where k is the number of factors or variables under investigation. Relative to a chosen origin X_1, X_2, . . . X_k, a regular simplex of edge length L is conveniently specified by the (k + 1) k design matrix D.

$$D = \begin{bmatrix} x_1 & x_2 & \cdots & x_k \\ x_1+p_2 & x_2+q_L & \cdots & x_k+q_L \\ x_1+q_2 & x_2+p_L & \cdots & x_k+q_L \\ \vdots & \vdots & & \vdots \\ x_1+q_L & x_2+q_L & \cdots & x_k+p_L \end{bmatrix}$$

where $p = (1/k\sqrt{2})\,[(k-1)+\sqrt{k+1}]$

and $q = (1/k\sqrt{2})\,[\sqrt{k+1} - 1]$

The values of p and q given here provide only one of an infinite number of orientations for the simplex design. However, Splendley shows that the procedure is relatively invariant to the design orientation. The rows of D give the k coordinates of the k + 1 vertices of the simplex. That is, the design points are the rows of D. The j^{th} row of D will be denoted vertorially by $\underline{d}_j$. The criterion for discarding any existing vector or row of D is the maximum current forecast error squared. This causes the system to move from a region of high forecast error to one of lower forecast error.

To apply this technique to a two-parameter or a three-parameter exponential smoothing model would require the following rules:

1. Let E(i) be the square of the forecast error at the i^{th} design point

$$E(i) = [F(i) - x(i)]^2, \quad i = 1, 2, \ldots N$$

Let the maximum value of E(i) occur at point $\underline{d}_j$. Form a new simplex by deleting $\underline{d}_j$ from D and substituting the new design point $\underline{d}_j^*$ where

$$d_j'^* = 2/k\ (d_1' + d_2' + \ldots + d_j - 1' + d_j + 1' + d_N') - d_j'$$

Calculate the forecast for the next period using the smoothing parameters which are the elements of d_j^*.

2. Apply rule 1 unless a design point has occurred in N successive simplexes without being eliminated. Should this situation arise for the i th design point, discard E(i) and calculate the forecast for the next period using the smoothing parameters in $\underline{d}_i$. Then apply rule 1.

3. Should E(i) be the maximum forecast error square in the n^{th} simplex and E(i)* be the maximum forecast error square in the n + 1 st simplex do not return to the n^{th} design. Instead of oscillating, move from the n + 1 st design by discarding the second largest forecast error square.

The application of the rules given above results in a shift in the values of the control parameters at each period. This characteristic may tend to make the design too sensitive to random fluctuations in the time series and thereby lead to an unstable, inaccurate forecasting system. On the other hand, the factorial design dictates a parameter change only when a statistically significant need to change is shown. Thus, some filtering of the noise occurs with the factorial design. However, in the presence of a small amount of random noise, there may be little difference in the stabilities and accuracies of the two systems.

Winter's Model

To test the two techniques discussed previously, a fundamental exponential smoothing model due to Winter's (10) is utilized. The model hypothesizes that a time series may be viewed as being composed of permanent and random components. Furthermore, the permanent component can be decomposed generally into level, seasonality and trend factors.

Let the time series 0, 1, 2, . . . t, t + 1 be postulated and let x_t be a particular realization of the time series at time t. Assuming the forecast is made for one period ahead only, the equation for the leveling component is

$$S_t\ (x) = \alpha\ \left(\frac{x_t}{F_{t - L}} \right) + (1 - \alpha)\ [S_{t-1}(x) + R_{t-1}\ (x)]$$

Equation for the seasonal adjustment is

$$F_t = \beta\ \left(\frac{x_t}{S_t} \right) + (1 - \beta)\ F_{t-L}$$

and for the trend factor is

$$R_t\ (x) = \gamma\ [S_t\ (x) - S_{t-1}\ (x)] + (1 - \gamma)\ R_{t-1}\ (x)$$

where S_t (x) = estimate of the level component at time t

α = smoothing constant

β = smoothing constant

γ = smoothing constant

L = periodicity of the "season"

F_t = seasonality factor at time t.

$R_t(x)$ = trend at time t.

The smoothing constants α, β and γ have to be positive and less than unity but they are not all necessary equal.

CHAPTER III

DESCRIPTION OF THE TEST CONDITIONS

To test the performance of the two adaptive control techniques, data were generated artificially and the forecasting methods applied to each series using Monte Carlo simulation. The data were generated on a UNIVAC 1108 computer. Artificial time series were used because it is possible to model specific time series characteristics. If the characteristics of the time series are known, then one may draw general conclusions about the performance of the prediction models on series possessing these characteristics. On the other hand, if the characteristics are not known, then only restricted conclusions can be made about the particular series.

Eight time series were generated, each possessing certain characteristics. Each of these time series has a basic deterministic form, but random variation has been super-imposed to make the series more representative of actual industrial series. Only normally distributed random variables are used for the sake of simplicity. The random variable is generated with a mean of zero and a variance which is either low, fairly low, fairly high or high. The variance is said to be low when the ratio of the variance and the trend component is one-fourth. In the presence of a non-zero constant component, the ratio of the variance and the constant component defines the magnitude of the variance of the time series. Likewise, when the ratio is one-half, three-fourths and one and one-fourth, we say that the variance is fairly low, fairly high and high, respectively. Each of these time series arbitrarily set as being 200 time units in length.

To show that the random variates generated are normally and independently distributed with a mean equal to zero and a variance equal to a specified constant, a histogram was first constructed to provide a good over-all picture of the data. Then the mean and the variance of the 200 random numbers were computed. A hypothesis was set up to test that the mean of the random variates equals to zero against the alternative that the mean is not equal to zero. Another hypothesis was also set up for testing the hypothesis that the variance equals to the specified constant against an alternative that it is not. These hypotheses were tested using common tests of significance. Then a chi-square goodness of fit test was performed to show that the random variates are indeed normally distributed. Finally to complete the tests, a run test is used to examine the randomness of the data generated, to make sure that there is no trend or correlation in the data.

The remaining portion of this chapter will be devoted to the description of each of the eight time series that were generated. Their characteristics and the series values will be presented. Hereafter, the time series will be referred to by number only.

Description of the Time Series Used

Series 1

This series contains a constant component and a linear trend component with superimposed random noise. The form of the generator equation is

$$X(t) = A + Bt + RV$$

where X(t) is the series value at time t; A is the constant component; B is the trend per time period; and RV is the random component which is normally and independently distributed (NID). For this series A has a value of zero and B has a value of ten. Figure 4 shows a portion of this time series for RV $\sim$ NID (0, 2.5).

Series 2

This series is identical to series 1 except there is an impulse at time 100. The constant component and the trend component are identical to those of series 1. Figure 5 shows a portion of this series for RV $\sim$ NID (0, 2.5).

Series 3

Series 3 is essentially the same as series 1 except that there is a step function in the trend component at time period 100. The form of the generator equation is as follows:

$$X_1(t) = A + Bt + RV \qquad \text{for } t < 100$$
$$X_2(t) = A + Dt + RV, \qquad \text{for } t \geq 100$$

where X_1 (t) and X_2(t) are the time series values for time period 1 to 100 and 100 to 200 respectively. A is the constant component for both time series and it takes on the value of zero. B, and C are the trend component and they take on the value of 10 and 50 respectively. Figure 6 shows a portion of this time series for RV $\sim$ NID (0, 2.5).

Series 4

Series 4 is very similar to series 3. It is a combination of a step and a ramp function where the time series is constant for time period 1 to 100 at which point there is a step increase. After the increase, the time series follows a linear trend. The form of the generator equation is as follows:

$$X_1\ (t) = A + RV \qquad \text{for } t < 100$$

$$X_2\ (t) = ct + RV \qquad \text{for } t \geq 100$$

where A the constant component is 10 and the trend component C is 100. Figure 7 shows a portion of this time series for RV $\sim$ NID (0, 7.5).

Series 5

Series 5 represents a series exhibiting strong seasonal or periodic variation. To generate such a time series, the following equation was used

$$X(t) = B \sin\left(\frac{2\pi}{L}\right) + B \cos\left(\frac{2\pi}{L}\right) + RV$$

where X(t) is the series value at time t, B is the amptitude of the sinusoid, L is the period of the sinusoid and RV is the random component. In this series, B has a value of 10 and L has a value of 12. Figure 8 shows a portion of this series for RV $\sim$ NID (0, 5.0).

Series 6

Series 6 is a twelve-point sine wave with one harmonic. To generate such a time series, the following equation was used.

$$X(t) = B \sin \left(\frac{2\pi}{L} \right) + B \cos \left(\frac{2\pi}{L} \right)$$

$$+ C \sin \left(\frac{4\pi}{L} \right) + C \cos \left(\frac{4\pi}{L} \right) + RV$$

where X(t) is the series value at time t, B and C are the amptitude and L is the periodicity of the series. In this series, B has a value of 20 and C has a value of 25 and L is 12. Figure 9 shows a portion the time series for RV $\sim$ NID (0, 12.5).

Series 7

Series 7 is composed of trend and seasonal components. The generator equation is as follows:

$$X(t) = Bt + C \sin \left(\frac{2\pi}{L} \right) + D \cos \left(\frac{2\pi}{L} \right) + RV$$

where X(t) is the series value at time t. B is the trend component and C and D are the amptitude of the cyclic wave and L is the periodicity of the cycle. B, C and D all takes on a value of 10. Figure 10 shows a portion of this time series for RV $\sim$ NID (0, 5.0).

Series 8

This series is one that exhibits high degree of autocorrelation. The form of the generator equation is as follows:

$$X(t) = \sum_{I=0}^{10} RV\ (t + I)$$

where X(t) is the series value at time t and RV is the random component. Thus, each value of the time series is formed by summing eleven random variables.

For example

$$X(1) = RV(1) + RV(2) + \ldots\ldots + RV(11)$$

$$X(2) = RV(2) + RV(3) + \ldots\ldots + RV(12)$$

Figure 11 shows a portion of this time series for $RV \sim NID(0, 12.5)$.

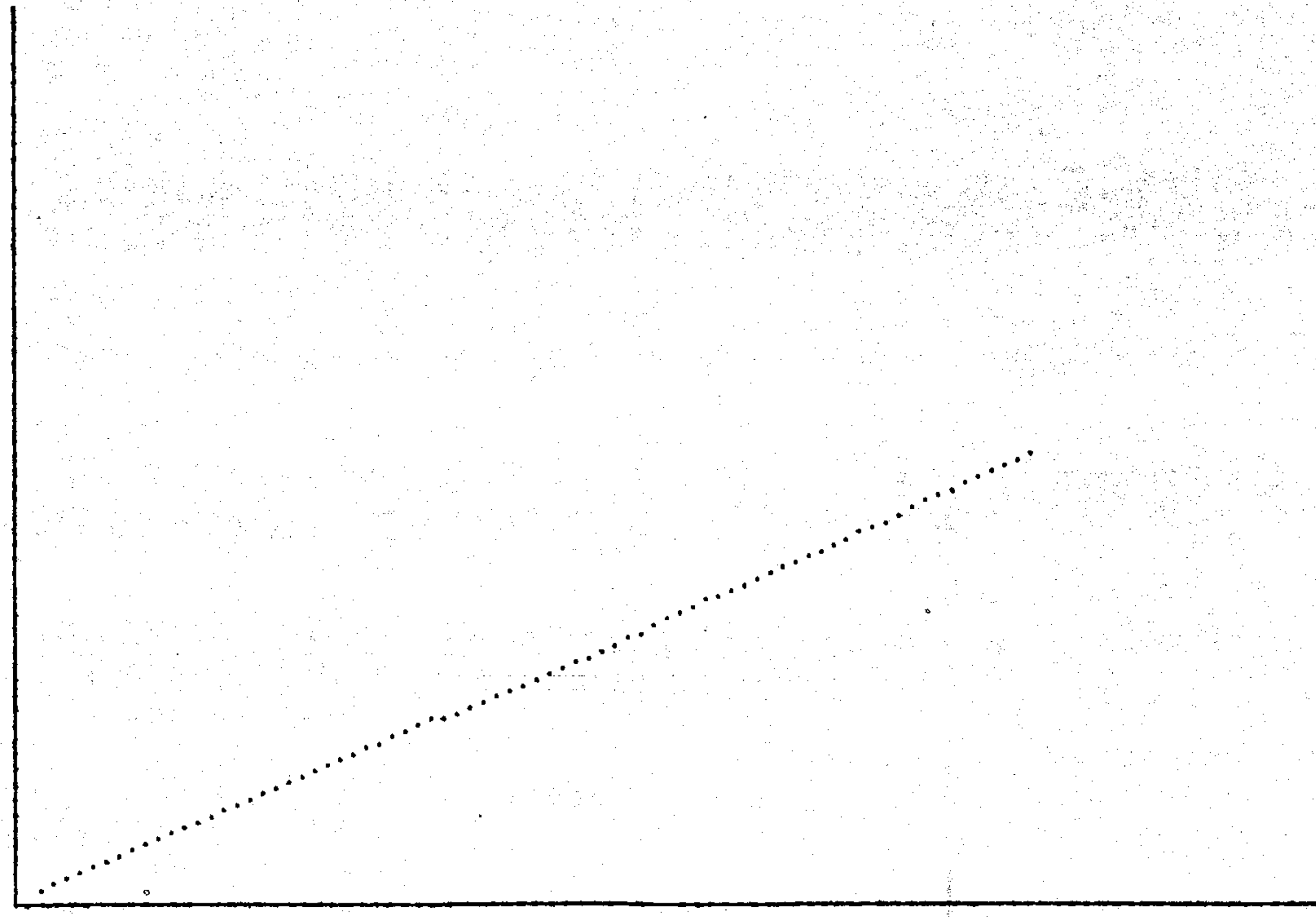


Figure 4. Time Series 1 -- Linear Trend with RV NID (0, 2.5).

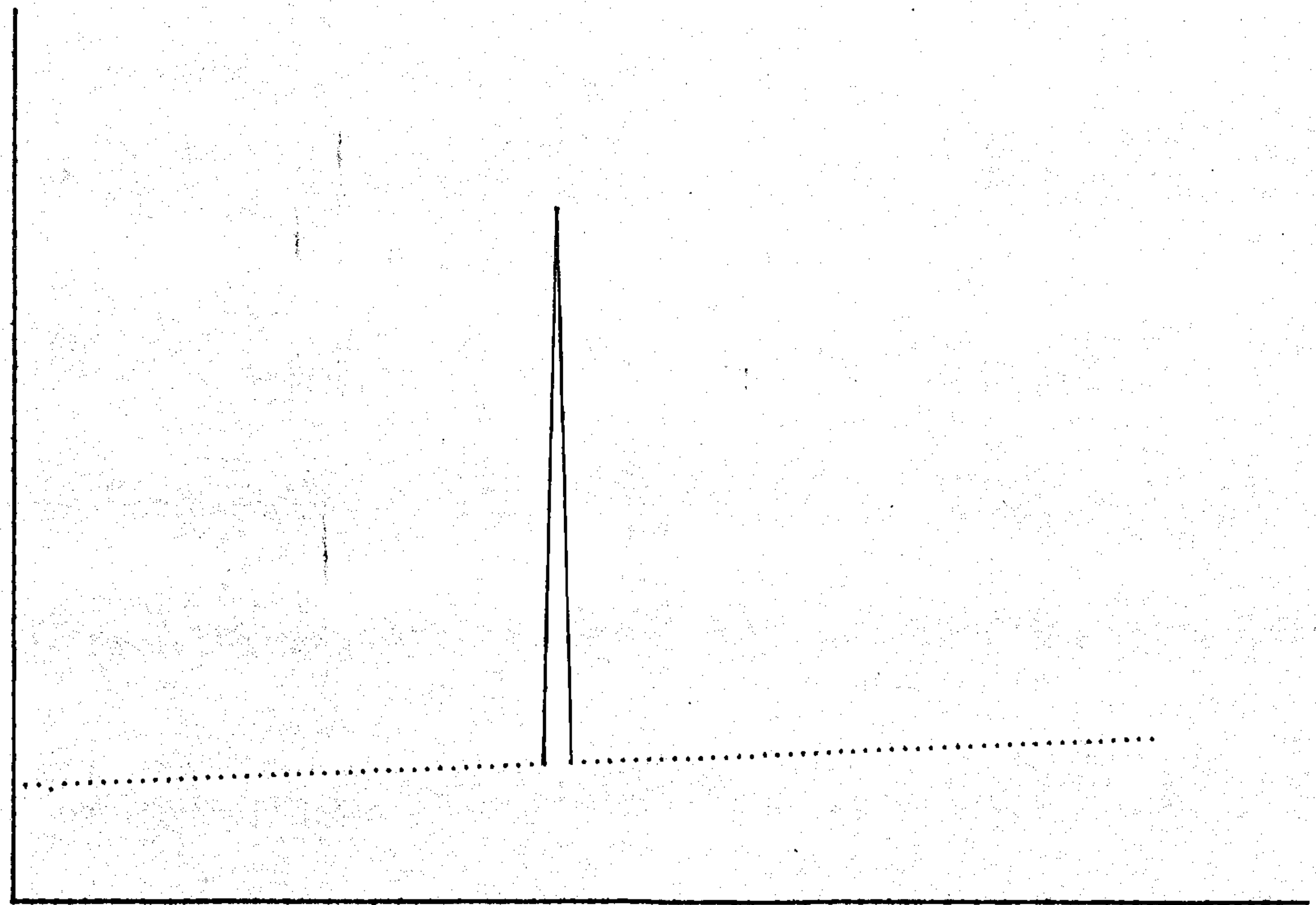


Figure 5. Time Series 2 -- Impulse with RV NID (0, 2.5).

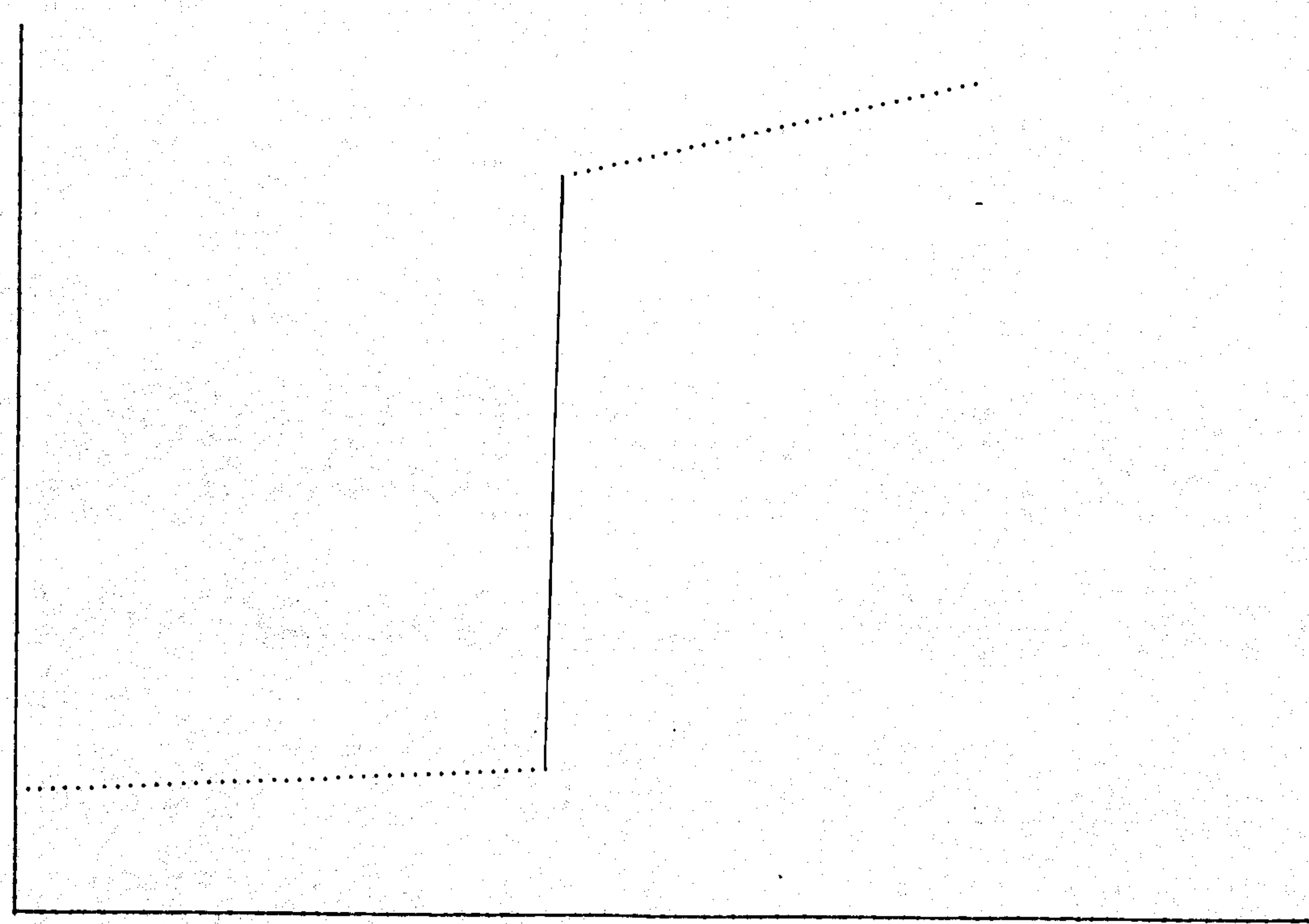


Figure 6. Time Series 3 -- A Combination of a Ramp and a Step with RV   NID (0, 2.5).

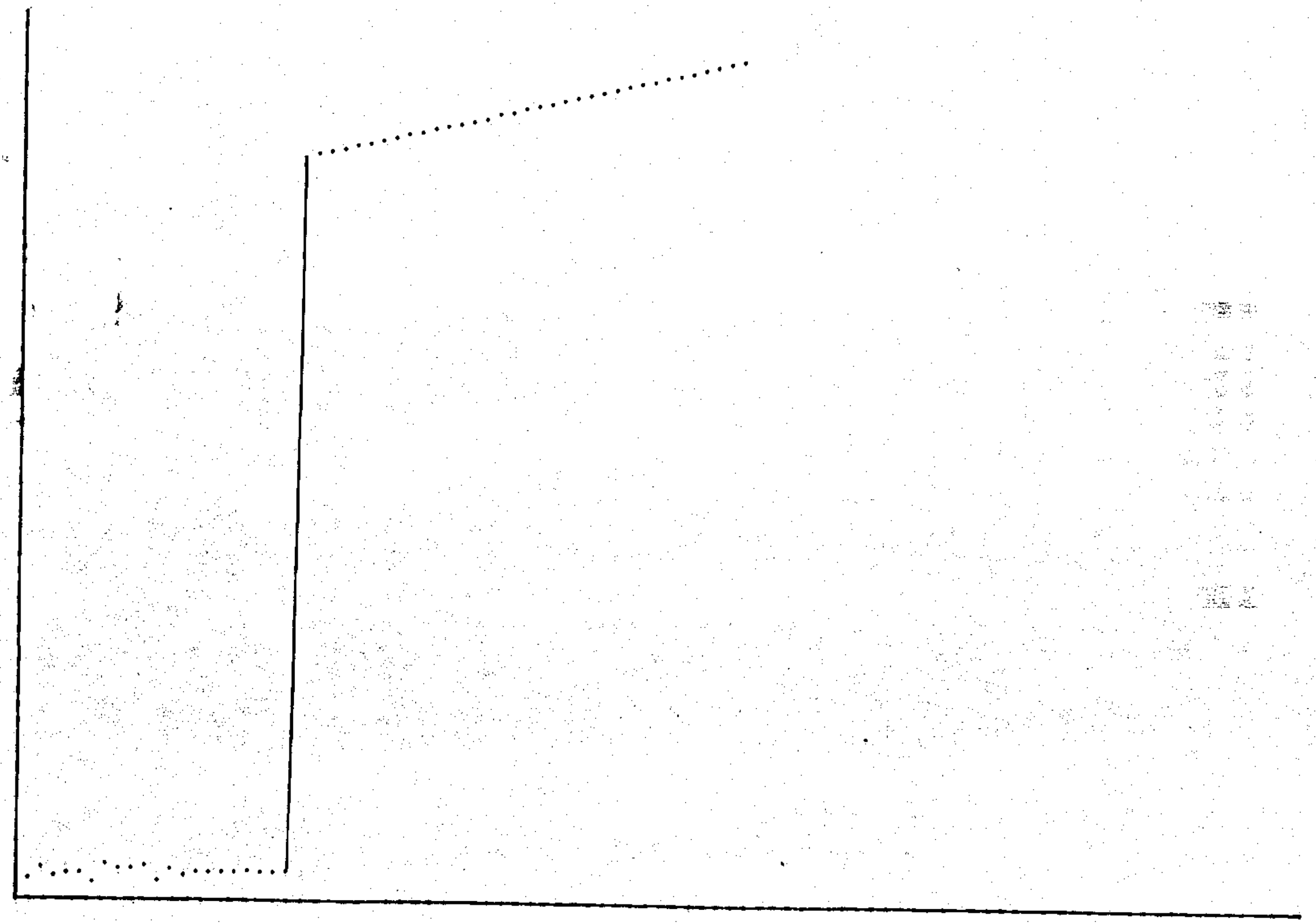


Figure 7. Time Series 4 -- A Combination of a Ramp and a Step with RV NID (0, 7.5).

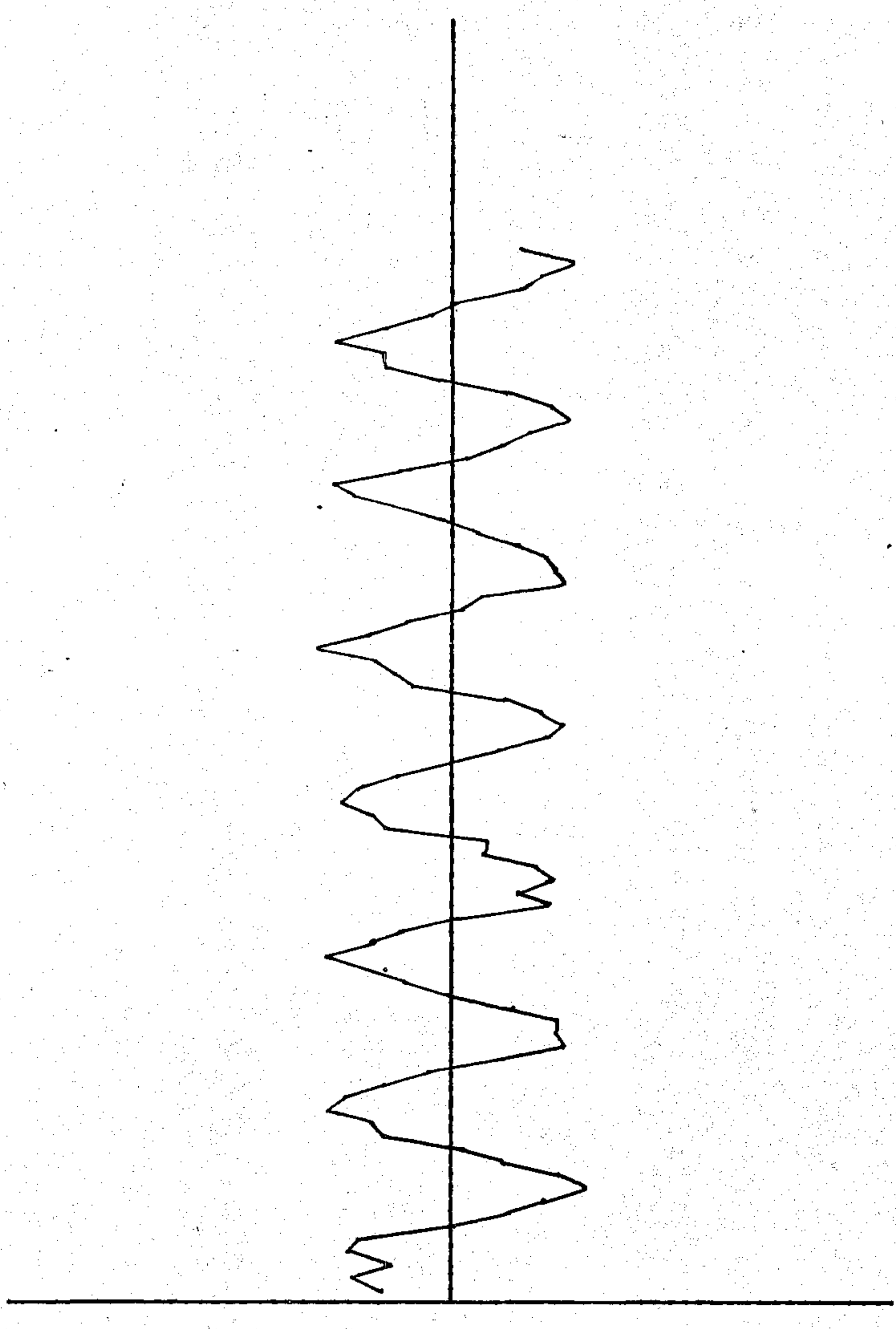


Figure 8. Time Series 5 -- Sinusoid with RV NID (0, 5.0).

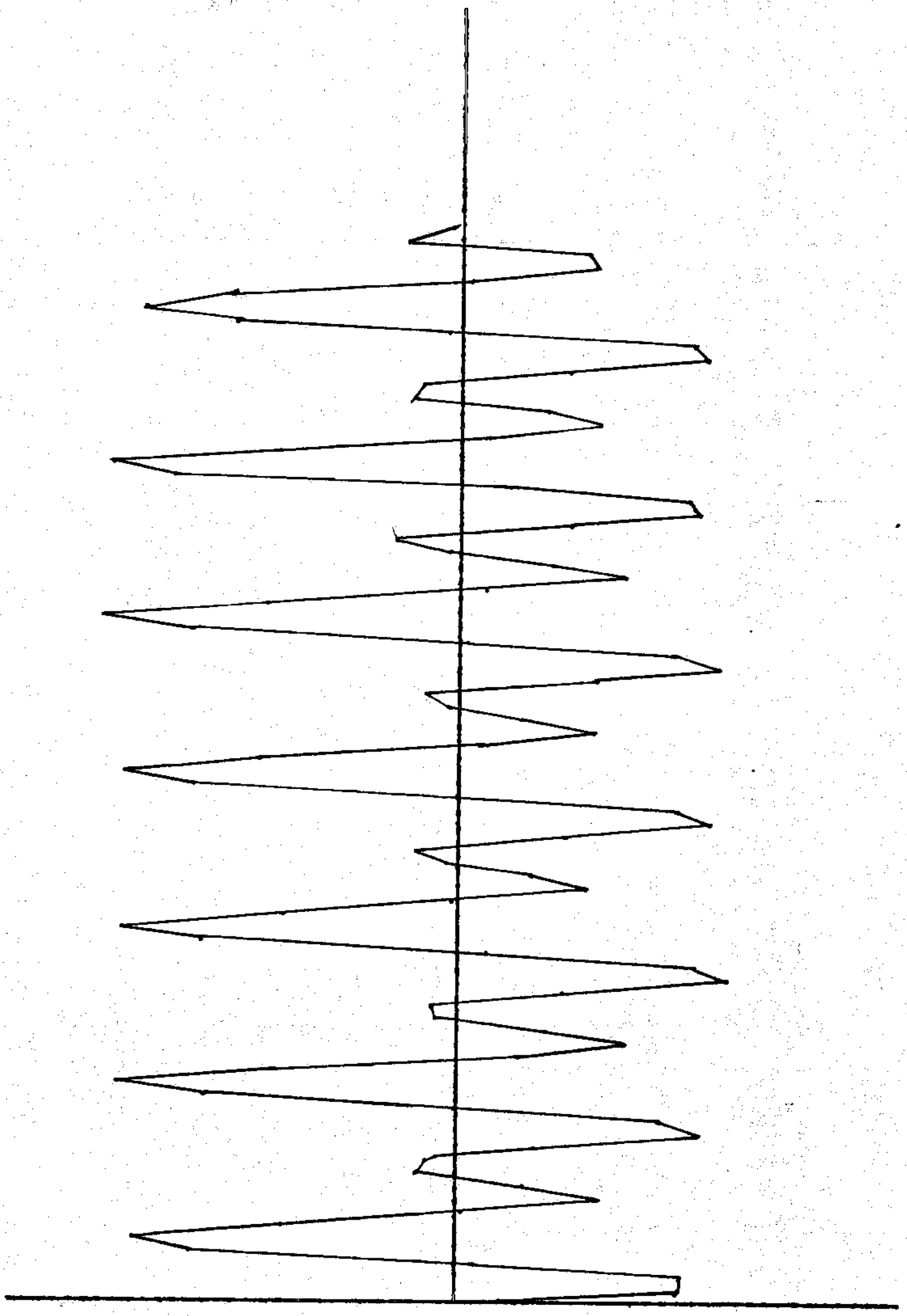


Figure 9. Time Series 6 -- Harmonic with RV NID (0, 12.5).

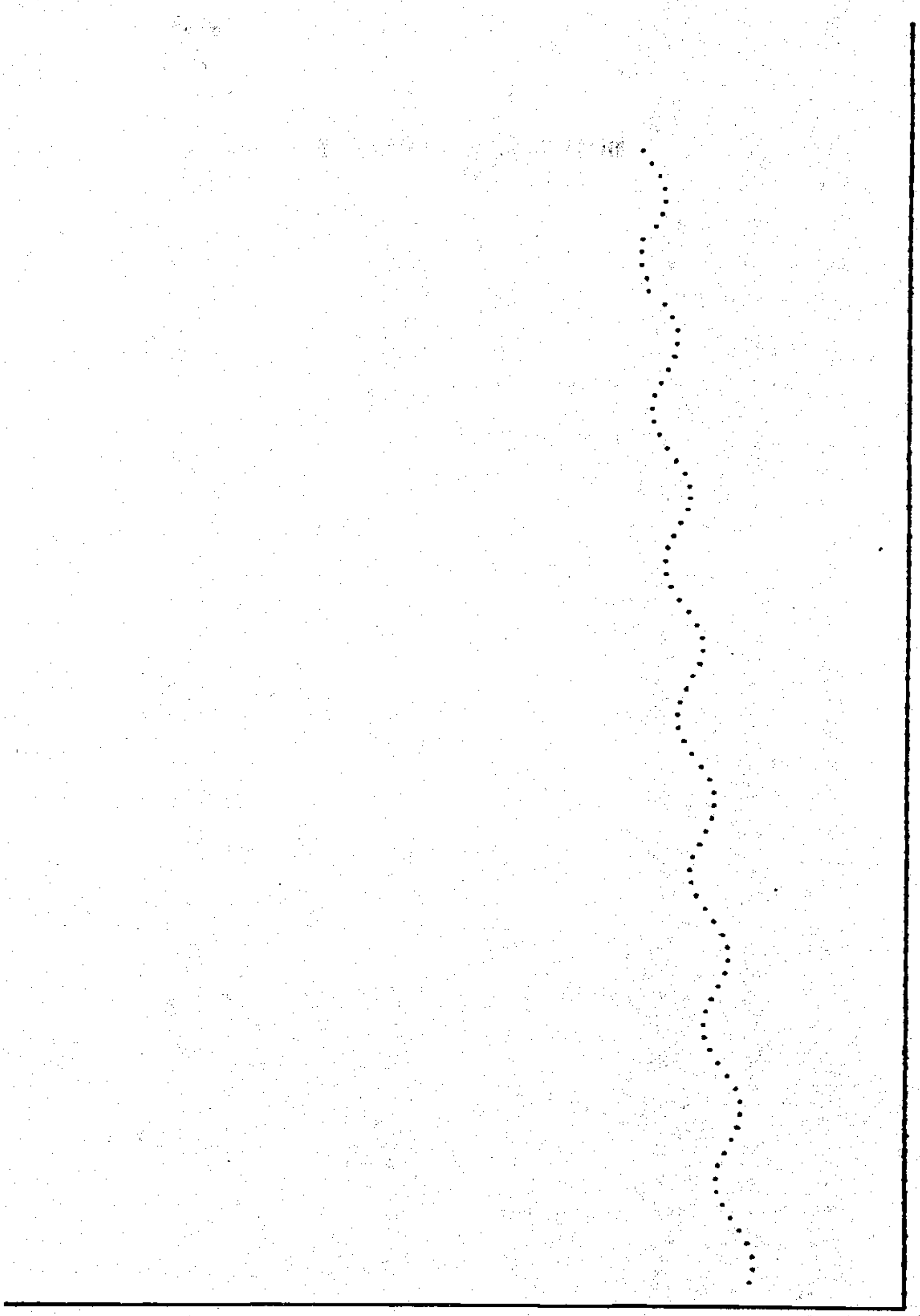


Figure 10. Time Series 7 -- Trend and Seasonal with RV NID (0. 5.0).

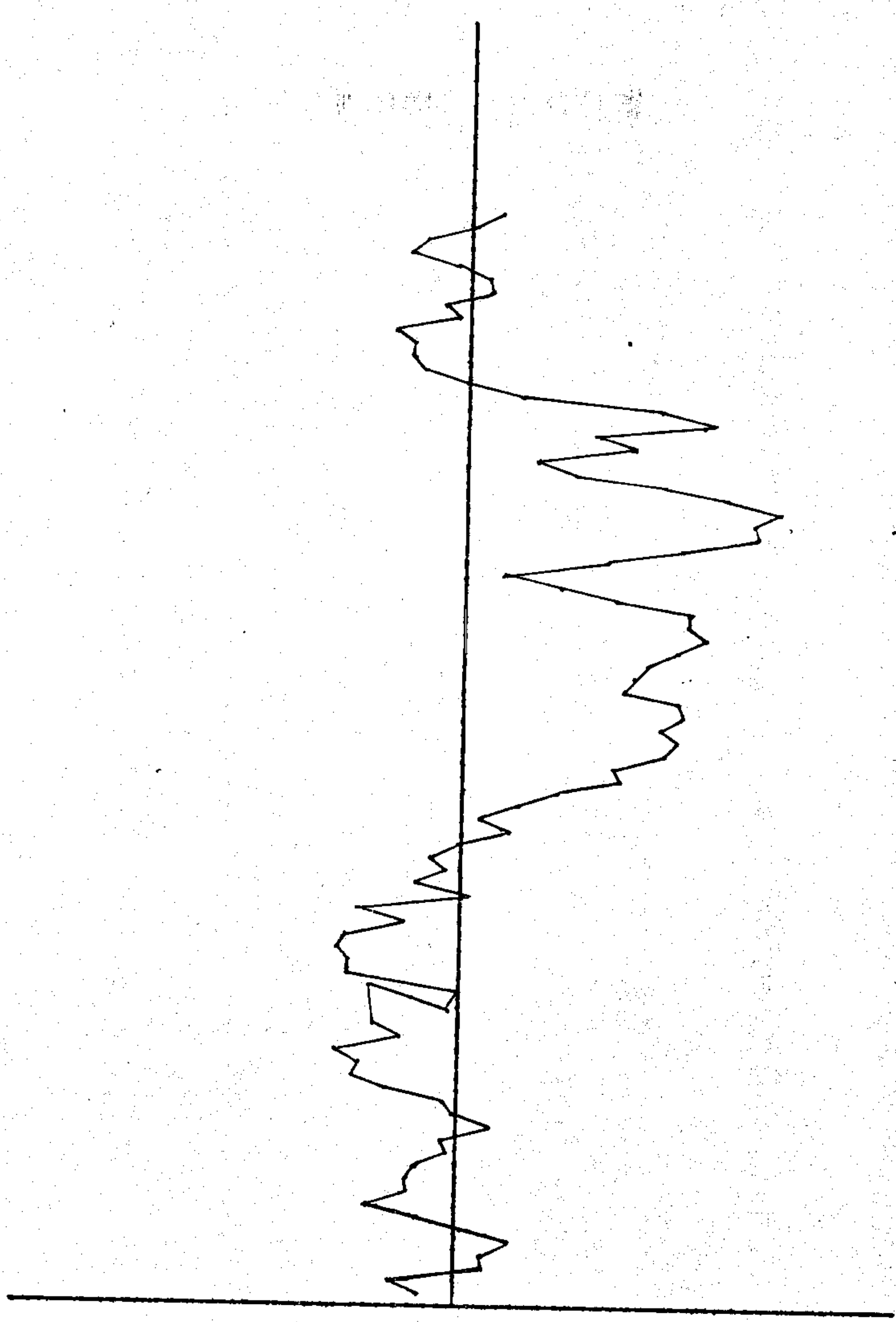


Figure 11. Time Series 8 -- Highly RV Autocorrelated Series with NID (0, 12.5).

CHAPTER IV

RESULTS AND CONCLUSIONS

This chapter presents the results obtained by applying Winter's model with the two EVOP forecasting techniques to the eight time series described in the previous chapter. An investigation of the sensitivity of the models to the initial values of the smoothing constants is presented first. An analysis of the sensitivity of the factorial EVOP to the spread of the experimental design and the sensitivity of the simplex EVOP to the edge length is then presented. Then the overall performance of the two forecasting models for the eight time series, each with four different magnitudes of error variance, is given. Finally, conclusions are given regarding the forecast accuracy and the rate of response of the two models.

Sensitivity to Initial Parameter Values

To investigate the sensitivity of the factorial EVOP to the initial values of the smoothing constants, different combinations of α, β and γ are used in series 1, 5 and 7. Series 1 is a two-parameter trend model, series 5 is a two-parameter seasonal model and series 7 is a three-parameter model which contains both trend and seasonality. Therefore, one series is used to represent each of the three classes of model. Each of these time series is run six times, each time using a different combination of α, β and γ. The random component of all three time series is arbitrarily selected to have a variance of 5.0. The results of this experiment is shown in Table 1.

Similarly, Table 2 exhibits the effect of initial values of α, β and γ on series 7, the effect of α and β on series 5, and the effect of α and γ on series 1.

Based on the results of Table 1 and Table 2, it is obvious that the simplex EVOP procedure is more sensitive to the initial values of the smoothing constants than the factorial EVOP procedure, and there is no specific pattern of sensitivity.

Sensitivity to Design Size

Another important question of concern is the sensitivity of the simplex procedure to the edge length utilized, and the sensitivity of the factorial procedure to the "spread" or upper and lower parameter limits of the experimental design.

To determine the proper spread for the factorial procedure, runs were made using spreads of 0.01, 0.03, 0.05 and 0.07. Here, spread is defined as the distance from the center point of the design to the face of the cube formed by the eight experimental points. Table 3 represents the crude sum of squares of the forecast errors resulting from these trials. From Table 3, it appears that series 1 and series 5 are not sensitive to the choice of the spread length.

From Table 3, it appears that not all time series are sensitive to the choice of the spread length. For series 1 and series 5 the choice of the spread length does not affect the crude sum of squared errors at all. Whereas, for series 2, 3, 4 and 6, a choice of 0.01 as a "spread value" seems to be the most appropriate. However, for

Table 1. Sum of Squares of Forecast Errors Obtained from Different Initial Smoothing Constants.

				Factorial EVOP					
α	β	γ	Series 7	α	β	Series 5	α	γ	Series 1
0.15	0.15	0.15	0.20816×10^8	0.15	0.15	0.32840×10^4	0.15	0.15	0.16046×10^8
0.05	0.05	0.05	0.21293×10^8	0.05	0.05	0.19420×10^4	0.05	0.05	0.16046×10^8
0.15	0.10	0.10	0.20851×10^8	0.15	0.10	0.17217×10^4	0.15	0.10	0.16046×10^8
0.15	0.10	0.15	0.20864×10^8	0.15	0.10	0.17217×10^4	0.15	0.15	0.16042×10^8
0.20	0.10	0.10	0.21066×10^8	0.20	0.10	0.22869×10^4	0.20	0.10	0.16042×10^8
0.20	0.15	0.10	0.20929×10^8	0.20	0.15	0.19607×10^4	0.20	0.10	0.16043×10^8

Table 2. Sum of Squares of Forecast Errors Obtained from Different Initial Smoothing Constants.

				Simplex EVOP					
α	β	γ	Series 7	α	β	Series 5	α	γ	Series 1
0.15	0.15	0.15	0.23104×10^6	.05	.05	0.10067×10^5	0.5	0.05	0.70119×10^5
0.05	0.05	0.05	0.56415×10^6	0.15	0.10	0.11675×10^7	0.15	0.10	*
0.15	0.10	0.10	0.36602×10^6	0.15	0.05	0.55444×10^8	0.15	0.05	0.55416×10^5
0.15	0.10	0.15	0.44029×10^6	0.20	0.10	0.63194×10^7	0.20	0.10	-
0.20	0.10	0.10	0.51913×10^6						
0.20	0.15	0.10	0.27436×10^6						

* Sum of Squares of Forecast Errors cannot be obtained because the System is Cycling Around a Single Point.

series 7 and 8 a choice of 0.05 as the "spread value" is appropriate.

A similar approach is used to determine the proper edge length for the simplex procedure. Again, runs using the edge length of 0.01, 0.03, 0.05 and 0.07 were made and Table 4 represents the crude sum of squared errors resulting from these trials. From Table 4, it appears that all time series are sensitive to the edge length. For series 1, 2, 3, 4 and 6, an edge length of 0.01 is the most appropriate. For series 5, 7 and 8 a larger value of the edge length is more appropriate

These values of edge length and spread were chosen arbitrarily for testing and are by no means optimal. They were chosen purely to give some relative indication of the sensitivity of the models to these variables. In actual practice, the best values of edge length or spread can be determined either by simulation and analysis of historical data or by trial and error experimentation in real time. Clearly the former method of setting these values would be the most desirable in an operating industrial prediction and control system.

Comparison of the Factorial and Simplex EVOP Procedures

Table 5 through 8 show the crude sums of squares of the forecast errors for all eight time series, with four different magnitudes of error variance, using both the factorial and the simplex EVOP techniques. The edge length and the spread for all runs were both chosen to be 0.05, which seems to work reasonably well for both methods. Also, both Roberts and Reed as well as Montgomery have suggested this value,

Table 3. Crude Sum of Squared Errors for the Eight Time Series Using Various Spread Values of the Factorial Design.

Time Series	Spread Values 0.01	0.03	0.05	0.07
1	0.20961×10^8	0.20842×10^8	0.20870×10^8	0.21172×10^8
2	0.26973×10^5	0.40215×10^5	0.49115×10^5	0.49277×10^5
3	0.28280×10^5	0.30866×10^5	0.31166×10^5	0.10994×10^6
4	0.65702×10^4	0.52514×10^6	0.64770×10^6	0.53834×10^7
5	0.16061×10^8	0.16054×10^8	0.16051×10^8	0.16051×10^8
6	0.13721×10^9	0.18435×10^9	0.22337×10^9	0.22142×10^9
7	0.24161×10^9	0.13778×10^9	0.12580×10^9	0.12582×10^9
8	0.29595×10^9	0.16723×10^9	0.17080×10^8	0.13885×10^9

Table 4. Crude Sum of Squared Errors for the Eight Time Series Using Various Edge Lengths of the Simplex Design.

Time Series	Edge Lengths 0.01	0.03	0.05	0.07
1	0.74698×10^6	0.96214×10^6	0.10509×10^7	0.71795×10^6
2	0.30221×10^5	0.61824×10^5	0.61212×10^5	0.67478×10^5
3	0.17372×10^5	0.22372×10^5	0.84950×10^7	0.12191×10^6
4	0.35501×10^4	0.19653×10^6	0.17618×10^8	0.32816×10^8
5	0.75523×10^5	Δ	0.58799×10^5	0.58227×10^5
6	0.11524×10^9	0.12876×10^9	0.17407×10^9	0.17455×10^9
7	0.51179×10^9	0.26339×10^9	0.51717×10^9	0.39536×10^9
8	0.43205×10^8	0.26454×10^8	0.11591×10^8	0.23889×10^8

$^{\Delta}$Note: The crude sum of squared forecast errors cannot be obtained because the system is cycling around a single point.

and it seems to be used in practice. The initial values of α, β, γ used were all arbitrarily chosen to be 0.1.

From Tables 5 through 8, we see that the factorial EVOP is better for certain time series, while the simplex EVOP is better for other time series. However, to further illustrate from a graphical point of view the performance of these two forecasting procedures, an inspection and analysis of Figures 12 through 21 was made. These graphs show portion of both the time series values and the predicated values using the two different procedures.

Time series 1, whose random variable is distributedN(0, 5.0), is shown in Figure 12 along with the results of the two prediction procedures. Series 1 exhibits a linear trend. According to the result in Table 6, the simplex procedure performs better than the factorial EVOP procedure. Most of the difference in the sum of square of forecast error between the two procedures can be accounted for by their performances at the first sixteen time periods. As shown in Figure 2, the factorial EVOP procedure fluctuates up and down resulting in large forecast error at the initial stage, but it finally stabilizes itself and follows the trend equally well as the simplex after the sixteen time periods have gone by.

Time series 2 is time series 1 with an impulse. The series selected for illustration also has a random variable distributed N(0, 5.0). Figure 13 shows the response of the two forecasting procedures to an impulse in the time series. As expected, the performances of both procedures were the same at the initial stages, before the impulse. At the point of impulse, the rate of response to the signal is much better in the case of factorial EVOP procedure. The effect of the impulse lasted for only a short time, and the factorial EVOP forecasting system stabilized quickly. However, the impulse effect lasted for a long time in the simplex EVOP forecasting system and the system stabilized itself gradually.

Time series 3 contains a step increase. For responses to a step increase, the results are quite different. They are indicated in Figure 14. The simplex EVOP seemed to "overshoot" and oscillate badly. The factorial EVOP appeared to perform much better than the simplex in the case of a step function.

Time series 4 also contains a step increase. The only difference between series 3 and 4 is that the time series before the signal input is a trend for series 3 while it is a constant for series 4. After the signal input, both time series takes on a trend. The responses to series 4 are very different than those of series 3. Here, the factorial EVOP seemed to "undershoot" first and then "overshoot" and very slowly oscillates to the new level and follows the trend. The simplex EVOP seemed to "overshoot" first but it reached the new level quickly without large error fluctuations. The results are indicated in Figure 15. The reason for this may be explained simply. In the simplex procedure, the control parameters adapt themselves each period to changes in the time series. Thus, it is faster for the simplex procedure to change its parameter and adapt to a new level when a drastic change from a constant level to that of a trend is encountered.

However, in the factorial EVOP procedure, when there is a change in level after the step impulse, several periods are then required to re-evaluate the range estimate of the standard deviation so that an upward shift can occur in order to adapt to the new trend.

Series 5 contains a basic sinusoid or periodic pattern and clearly demonstrated the superiority of factorial EVOP procedure. In Figure 16, it can be seen that the simplex EVOP procedure resulted in very large prediction errors because it is out of phase with the signal. The factorial EVOP procedure, with its statistical control limits, allows parameter modifications which result in a predicted series which follows closely the periodic behavior of the series.

Series 6 is a sine wave with one harmonic. In Figure 17 this harmonic series has a random error component which is distributed N (0, 2.5) superimposed on it. It can be seen that the simplex EVOP procedure performed slightly better than the factorial EVOP. Nevertheless, when the random variable is distributed N (0, 12.5), that is when the system is noisy, a reversal occurred. The factorial EVOP procedure performed better when the variance of the random component is high, and the result is as shown in Figure 18. It seems that when the system is noisy, the continuous parameter changes inherent in the simplex EVOP procedure introduce additional error in series exhibiting periodic behavior.

Series 7 contains both trend and periodic components. As shown in Figure 19, the simplex EVOP procedure performs better than the factorial EVOP procedure. This is another case which confirms the superiority of simplex EVOP in the presence of a trend component.

Series 8, which is very highly auto-correlated, is shown in Figure 20 along with the results of the two prediction procedures. As expected, the factorial EVOP procedure performs better than the simplex EVOP since the high auto-correlation structure produces a time series that appears to be cyclic.

In conclusion, we can state that simplex EVOP procedure yields more accurate results in the presence of a trend component. On the other hand, for time series that shows periodic behavior, the factorial EVOP procedure proved significantly better than the simplex EVOP procedure. Finally, it can also be generally concluded that the factorial EVOP procedure gives better responses to standard signal inputs such as the impulse and the step functions. However, when there is a drastic change in the level of the time series before and after the signal input, the simplex EVOP procedure will give a better rate of response.

Table 5. Crude Sum of Squared Errors for the Eight Time Series Having RV NID (0, 2.5).

Time Series	Simplex EVOP	Factorial EVOP
1	0.59184×10^5	0.62400×10^7
2	0.15445×10^9	0.22742×10^9
3	0.36772×10^9	0.13373×10^9
4	0.11589×10^8	0.17082×10^8
5	0.35174×10^6	0.15630×10^5
6	0.94746×10^4	0.13763×10^5
7	0.38766×10^6	0.19011×10^8
8	0.19711×10^8	0.63847×10^5

Table 6. Crude Sum of Squared Errors for the Eight Time Series Having RV NID (0, 5.0).

Time Series	Simplex EVOP	Factorial EVOP
1	0.58799×10^5	0.16051×10^8
2	0.17470×10^9	0.22337×10^9
3	0.51717×10^9	0.12580×10^9
4	0.11591×10^8	0.17086×10^8
5	0.84950×10^7	0.31166×10^5
6	0.61212×10^5	0.49115×10^5
7	0.10509×10^7	0.20870×10^8
8	0.17618×10^8	0.64770×10^6

Table 7. Crude Sum of Squared Errors for the Eight Time Series Having RV NID (0, 7.5).

Time Series	Simplex EVOP	Factorial EVOP
1	0.58987×10^{5}	0.62509×10^{7}
2	0.11534×10^{9}	0.21533×10^{9}
3	0.41945×10^{9}	0.14291×10^{9}
4	0.11594×10^{8}	0.19553×10^{8}
5	0.17583×10^{6}	0.46579×10^{4}
6	0.39125×10^{8}	0.66577×10^{8}
7	0.52622×10^{6}	0.19022×10^{8}
8	0.25583×10^{9}	0.13605×10^{11}

Table 8. Crude Sum of Squared Errors for the Eight Time Series Having RV NID (0, 12.5).

Time Series	Simplex EVOP	Factorial EVOP
1	0.58612×10^5	0.12300×10^8
2	0.14142×10^9	0.20198×10^9
3	0.30577×10^9	0.13444×10^9
4	0.11598×10^8	0.17093×10^8
5	0.12370×10^7	0.68923×10^4
6	0.35904×10^8	0.17658×10^6
7	0.13488×10^6	0.12397×10^8
8	0.41784×10^8	0.30587×10^5

CHAPTER V

RECOMMENDATIONS FOR FURTHER RESEARCH

Two areas of further research are naturally suggested by the work done in this study. The first area involves the extension of the simplex EVOP procedure, while the second area involves a further detailed investigation of the effects of the basic nature of the time series to evolutionary operation procedures.

The first area of further research involves making the simplex more flexible in its use. In this thesis, the optimum operating conditions is obtained by evaluating the output from a system at a set of points forming a regular simplex in a factor space, and continually forming new simplexes by reflecting one point in the hyperplane of the remaining points. It is assumed that the relative steps to be made in varying the factors are known, and thus makes their strategy rather rigid for general use. Nelder and Mead (5) have developed a method of making the simplex adaptive to the local shape of the response surface. Comparison of this modified simplex method can be made with the regular simplex to see if it improves the forecast accuracy, and especially the rate of response to standard signals.

The second area of research is an extension of the work done for this thesis. From this study, it was shown that the simplex EVOP procedure performs well the trend model and the factorial EVOP performs well the seasonal model. It would be of great interest to investigate the effect of the simplex EVOP procedure on time series with varying magnitudes of the trend component. Similarly, it is possible to determine the effect of the factorial EVOP procedure on time series with varying magnitudes of the amplitude and the periodicity of the cycle. These investigations would be of value on further determining the characteristics of the two EVOP forecasting techniques.

APPENDIX

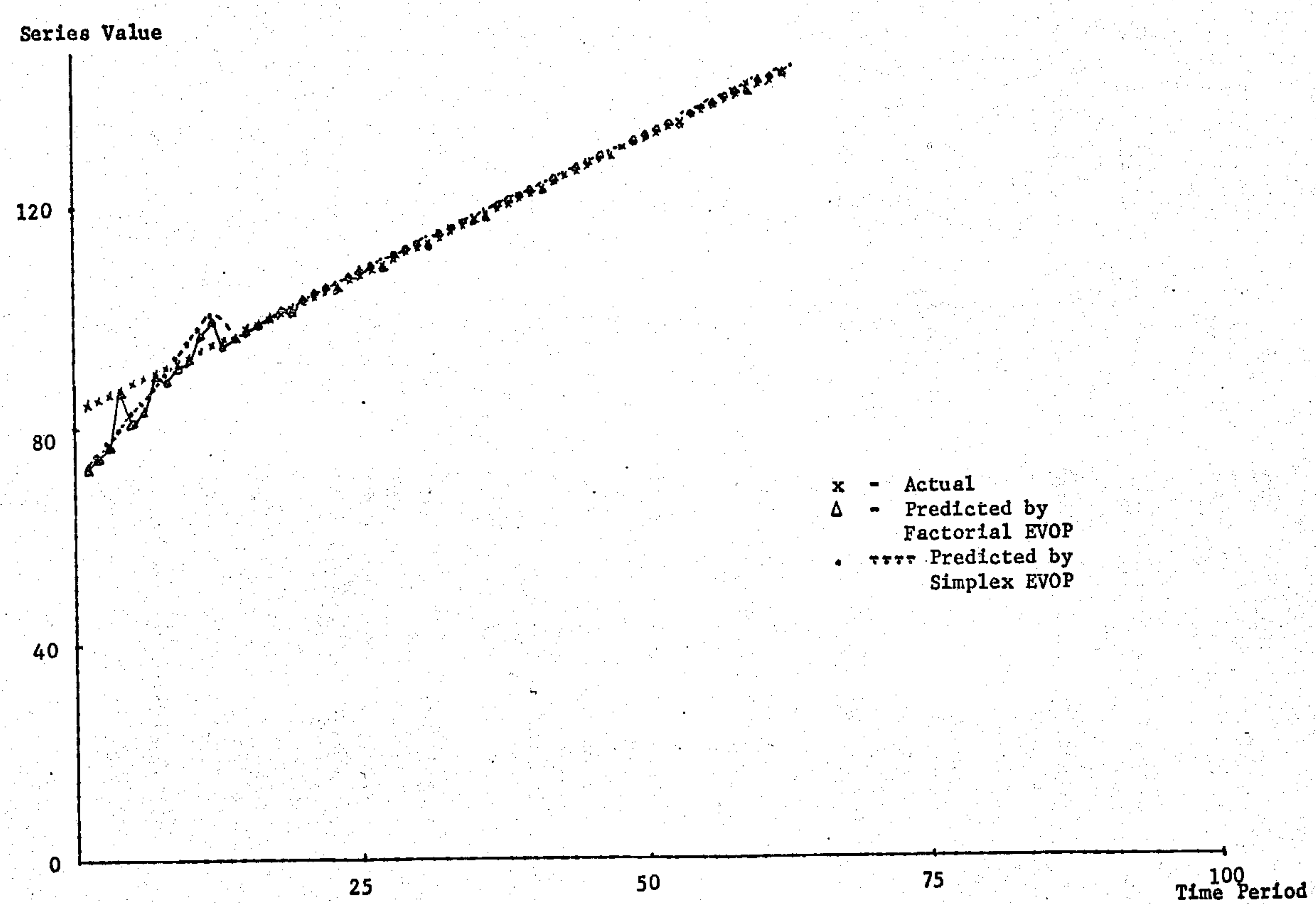


Figure 12. Performance of Factorial EVOP and Simplex EVOP on Series 1 with RV ~ NID (0, 5.0).

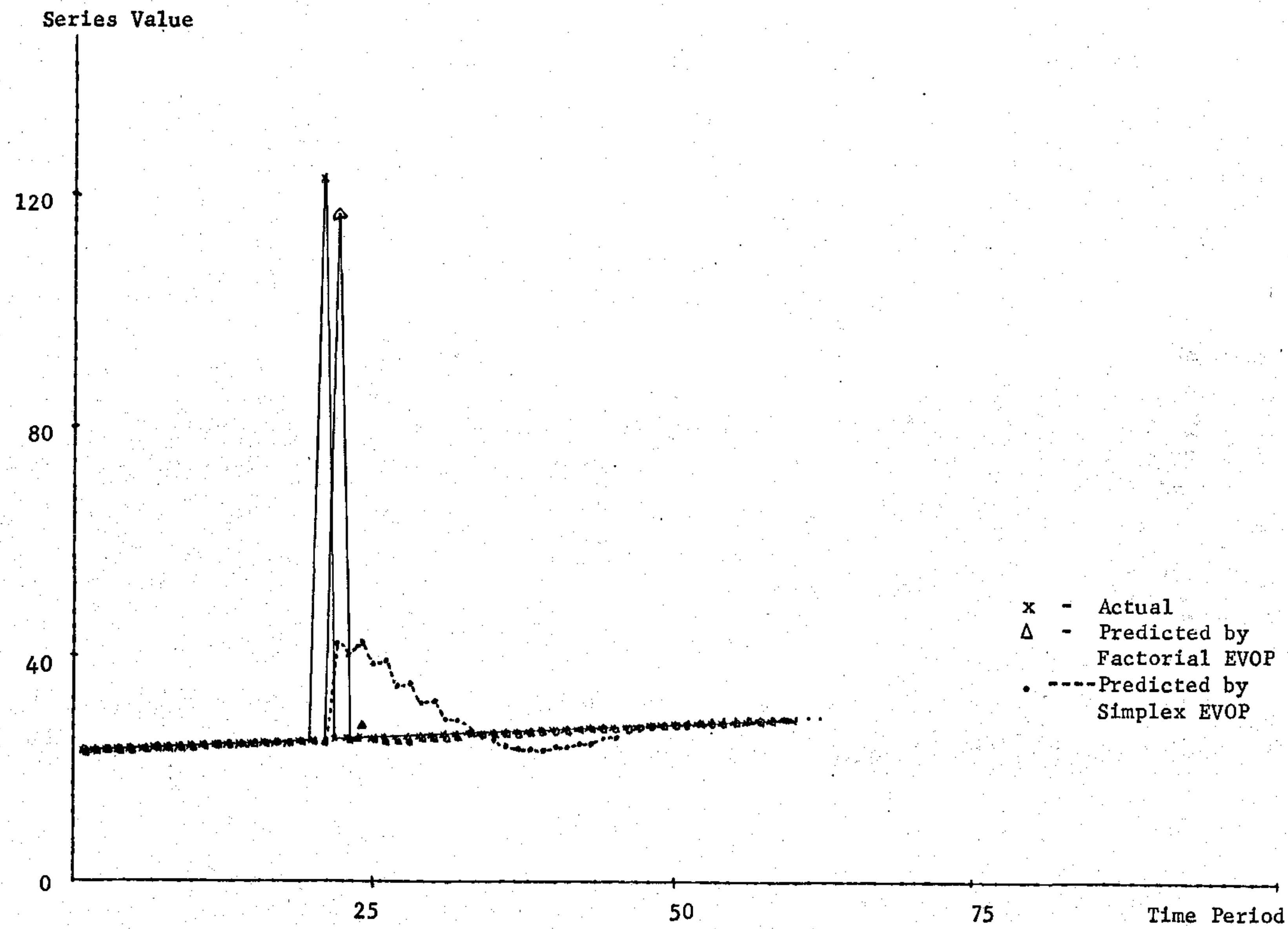


Figure 13. Performance of Factorial EVOP and Simplex EVOP on Series 2 with RV ~ NID (0, 5.0).

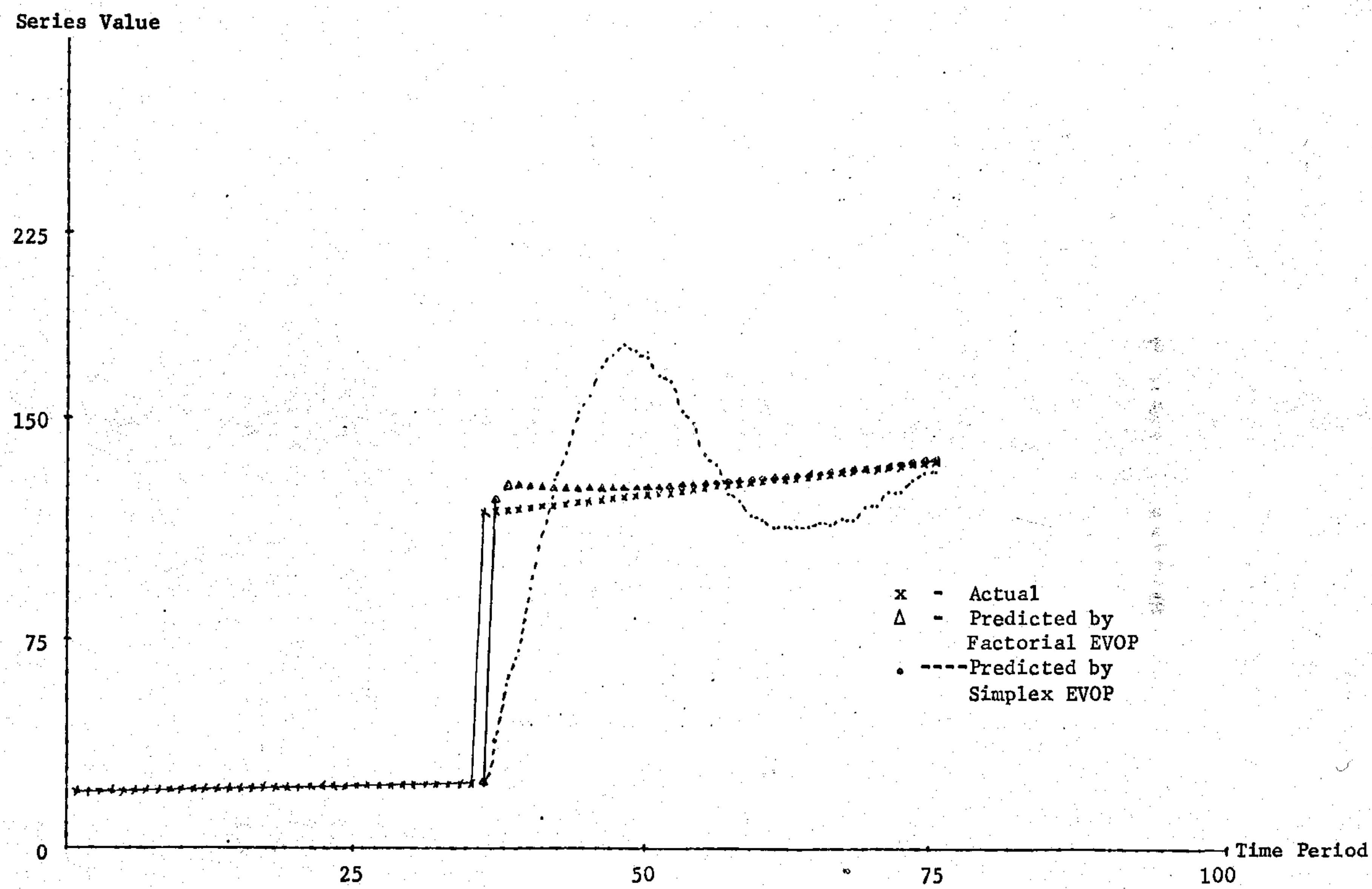


Figure 14. Performance of Factorial EVOP and Simplex EVOP on Series 3 with RV ~ NID (0, 5.0).

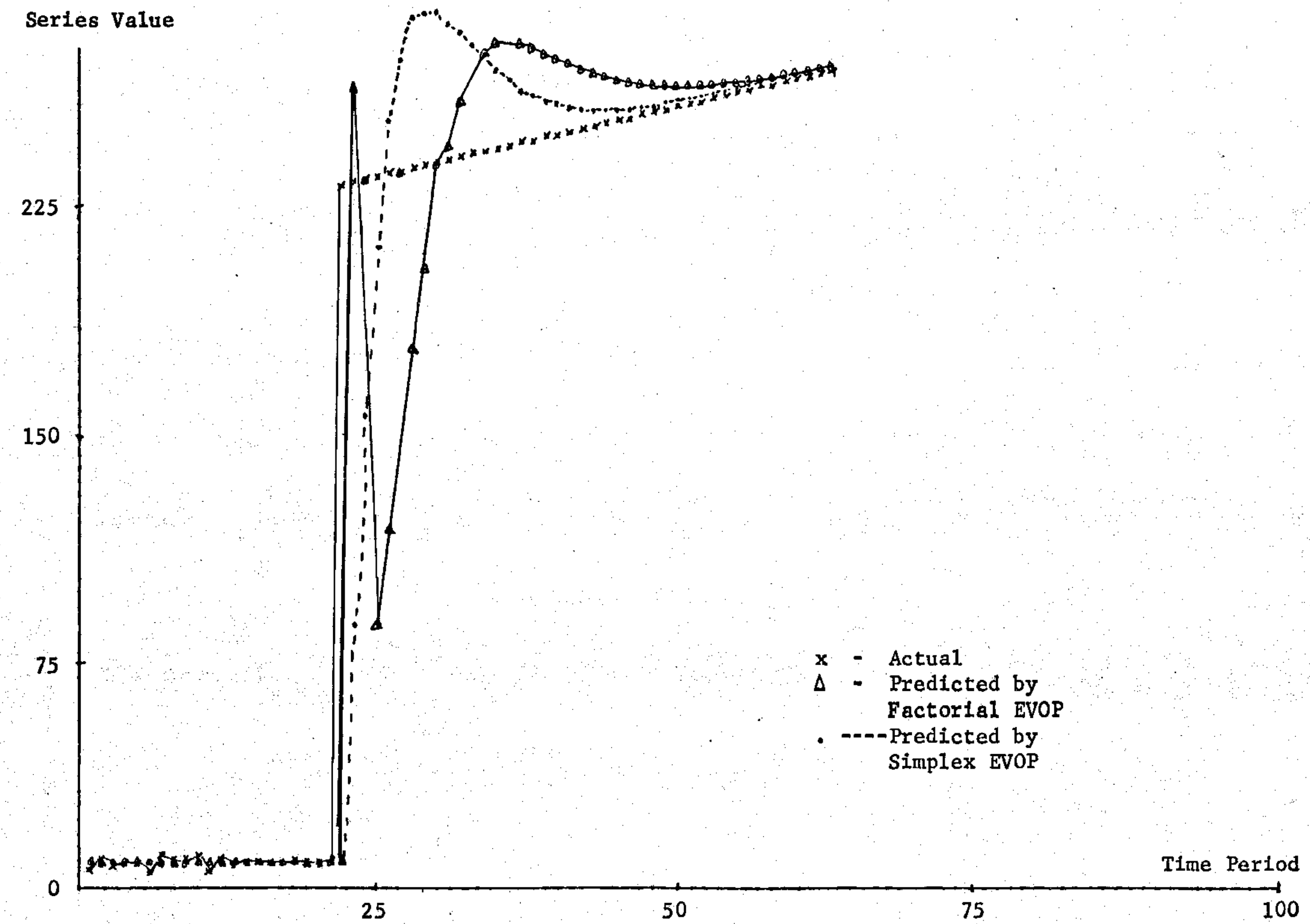


Figure 15. Performance of Factorial EVOP and Simplex EVOP on Series 4 with RV $\sim$ NID (0, 5.0).

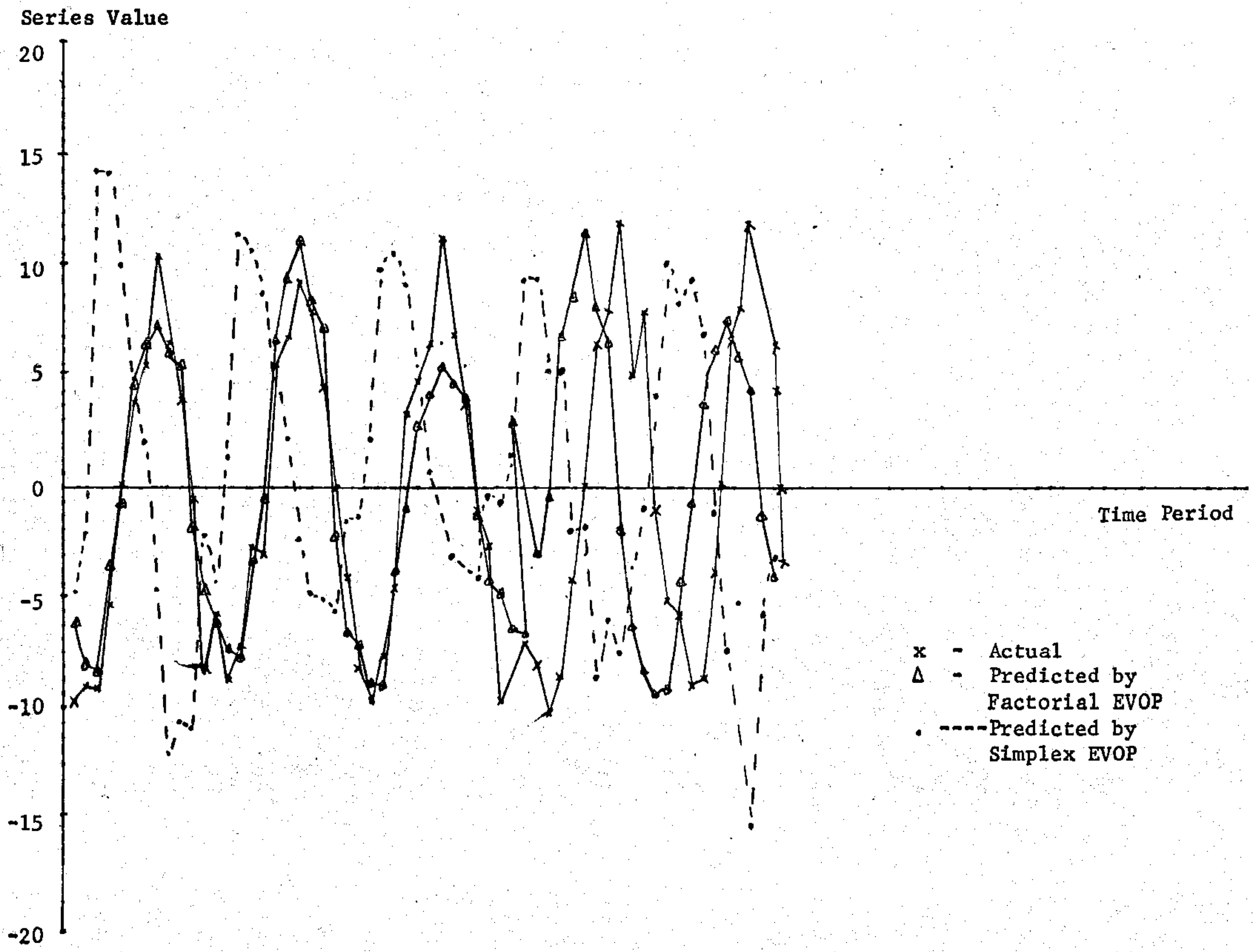


Figure 16. Performance of Factorial EVOP and Simplex EVOP on Series 5 with RV $\sim$ NID (0, 5.0).

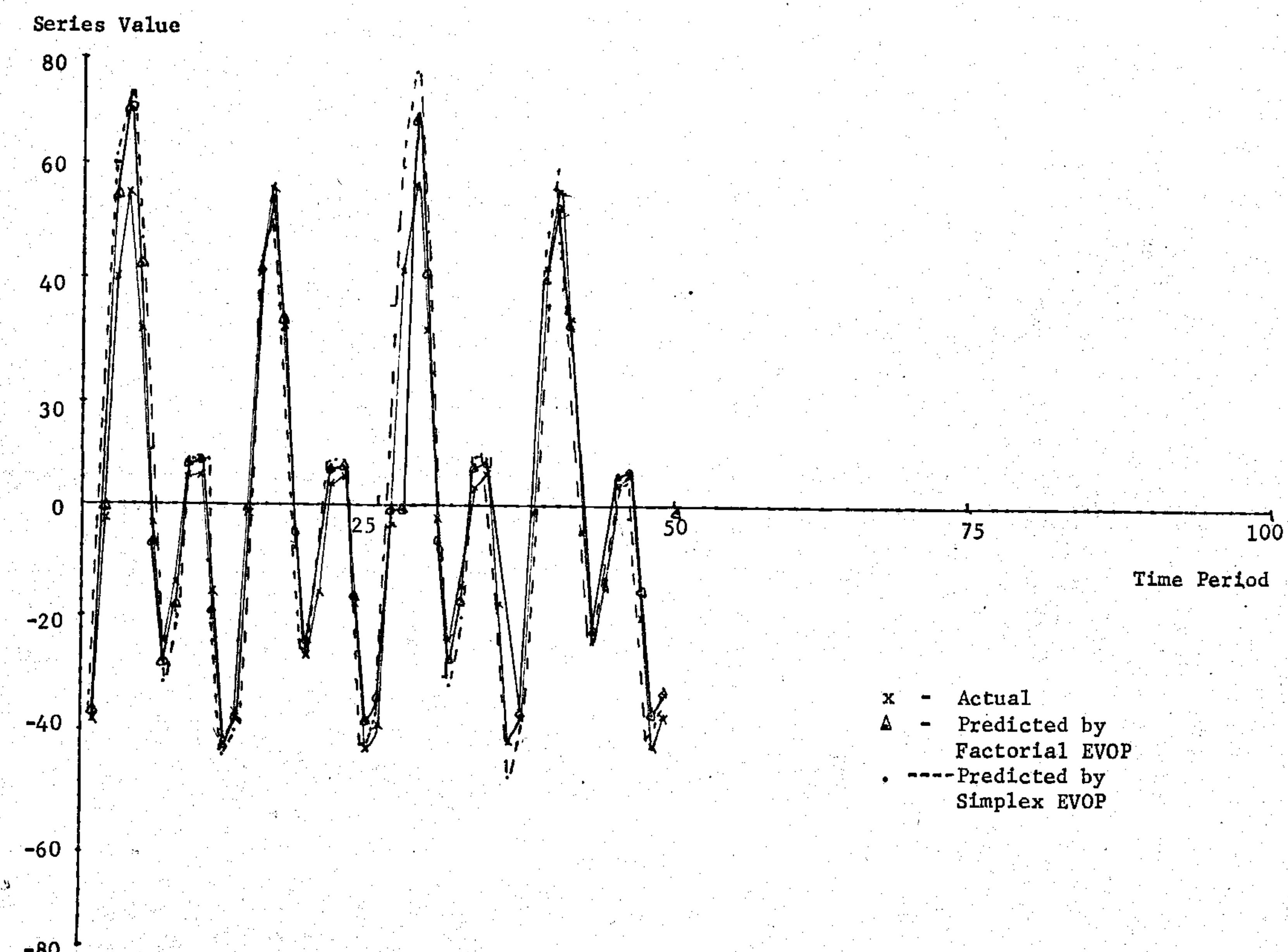


Figure 17. Performance of Factorial EVOP and Simplex EVOP on Series 6 with RV ~ NID (0, 2.5).

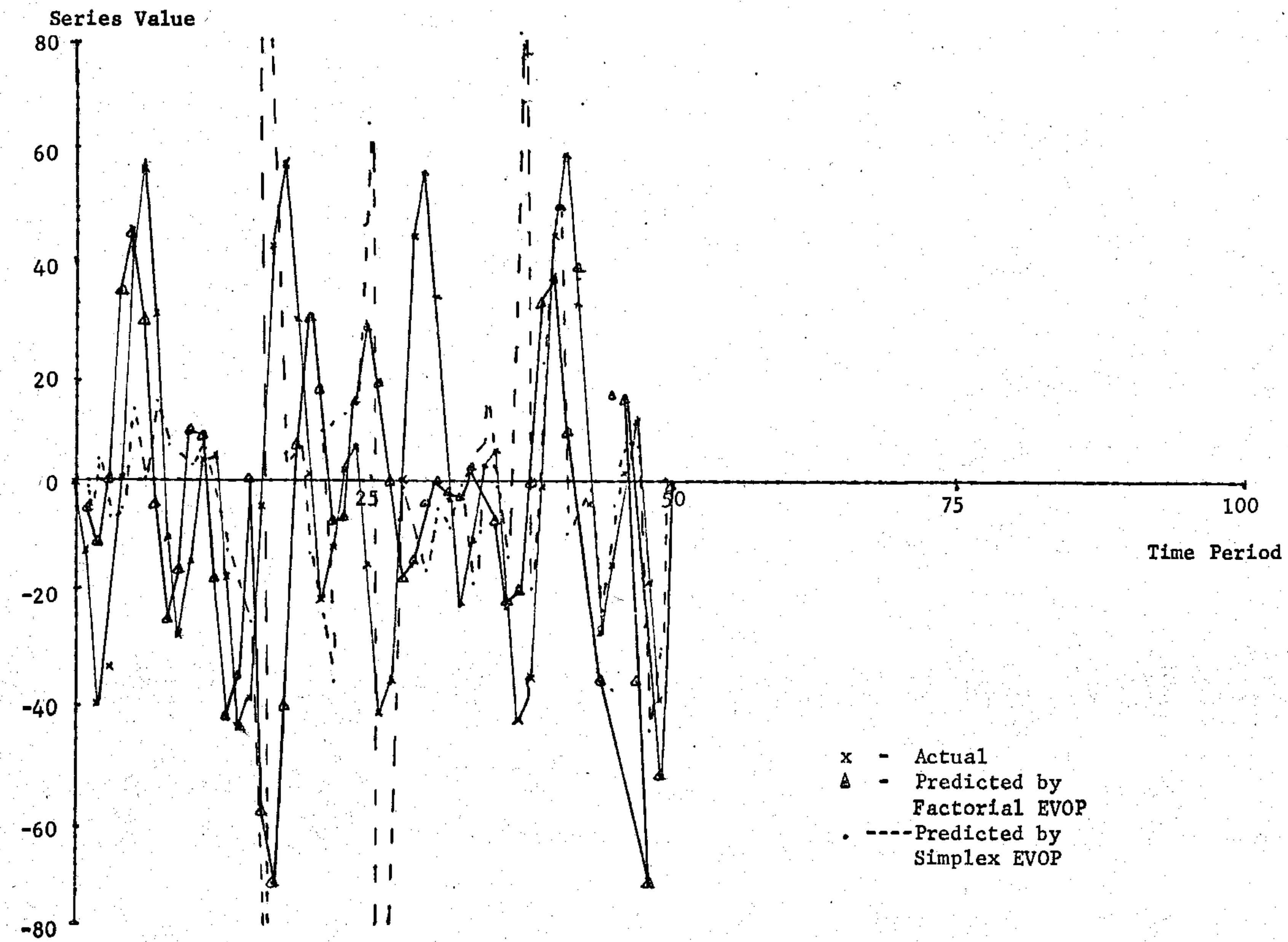


Figure 18. Performance of Factorial EVOP and Simplex EVOP on Series 6 with RV $\sim$ NID (0, 12.5).

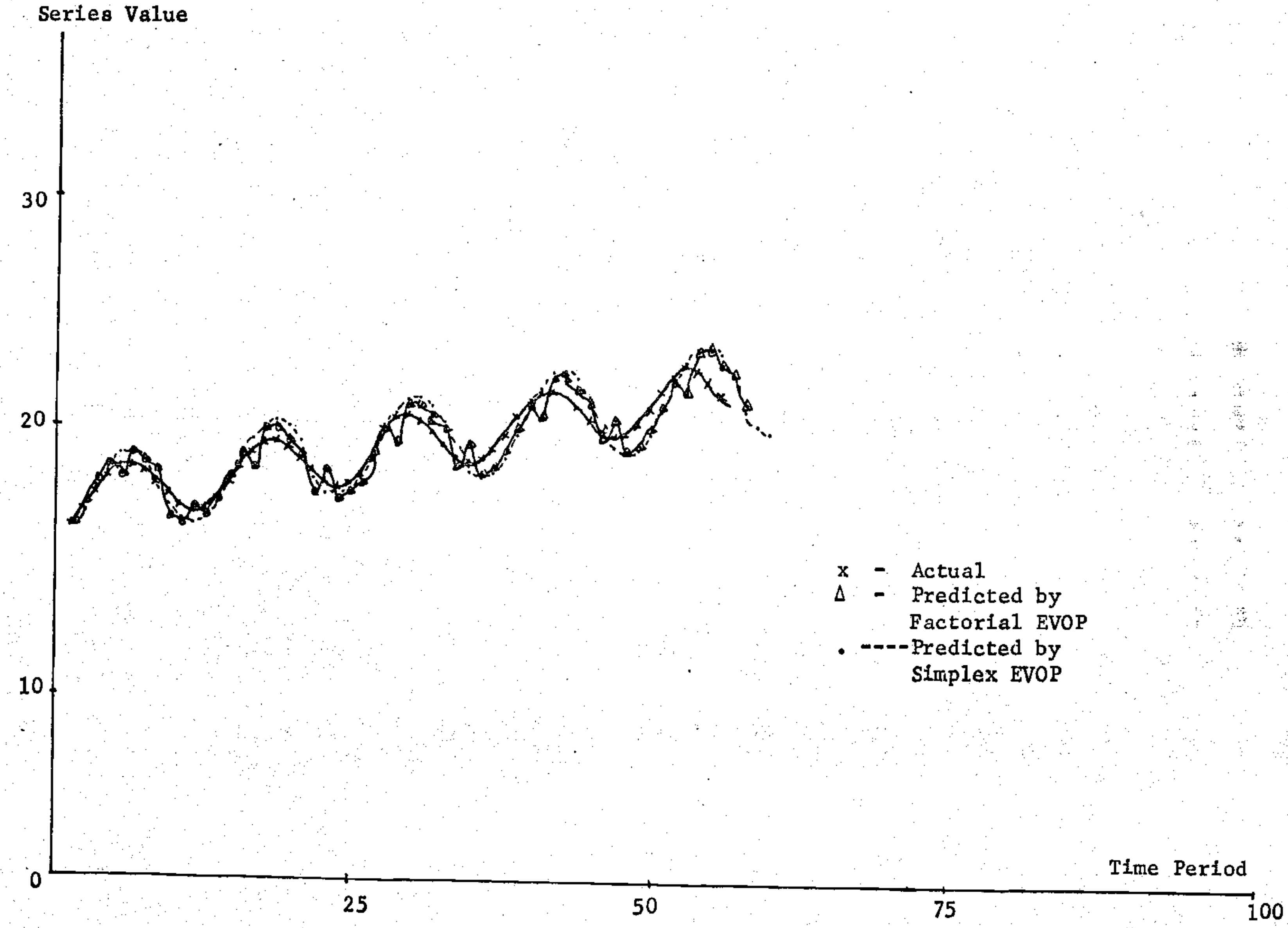


Figure 19. Performance of Factorial EVOP and Simplex EVOP on Series 7 with RV $\sim$ NID (0, 5.0).

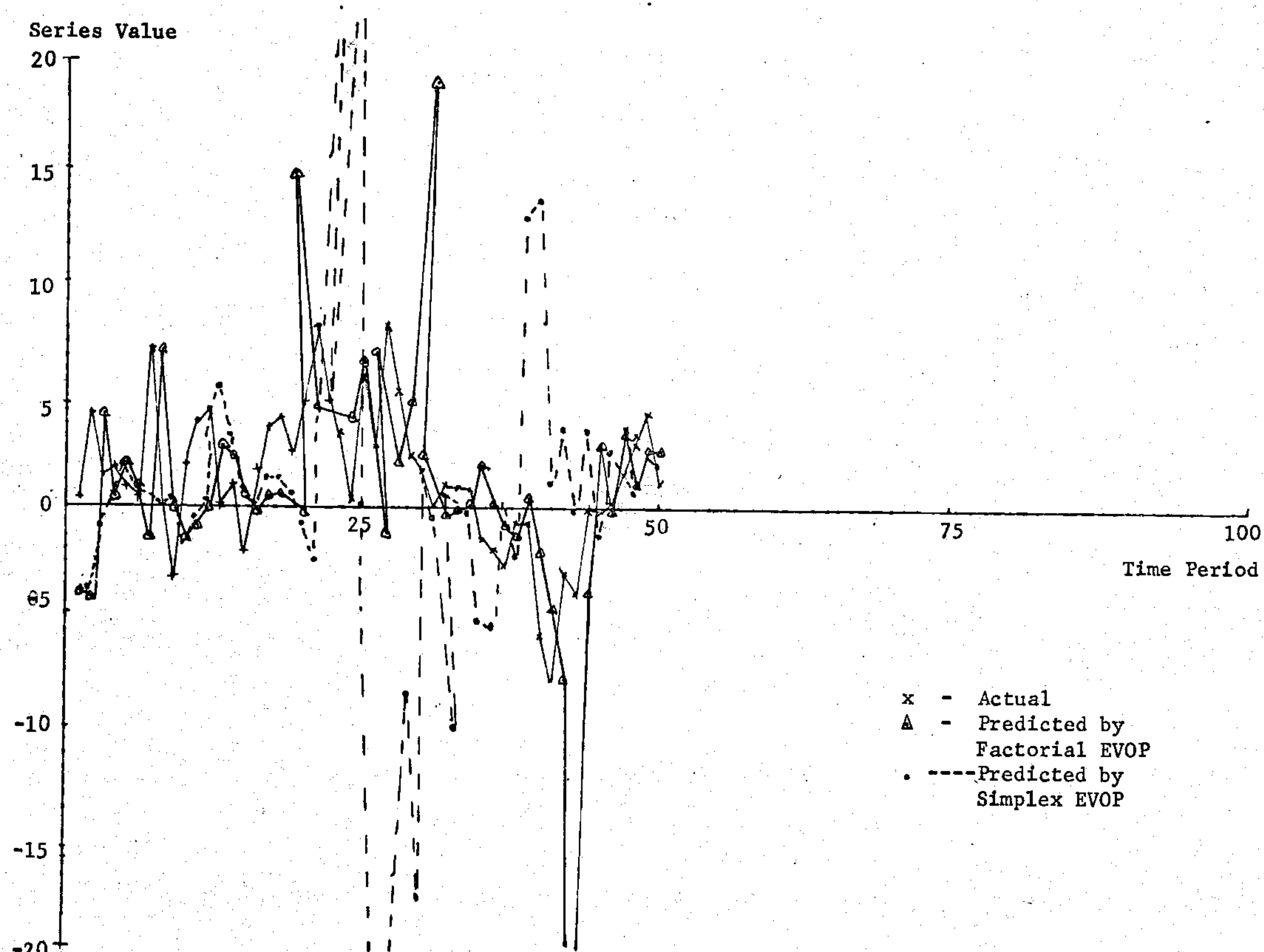


Figure 20. Performance of Factorial EVOP and Simplex EVOP on Series 8 with RV ~ NID (0, 5.0).

```
@FOR,IS MAIN
FOR S10A-02/27/73-13:31:46 (,0)

   MAIN PROGRAM

   STORAGE USED: CODE(1) 000024; DATA(0) 012742; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   BLOKA   000012
    0004   BLOKB   000033
    0005   BLOKC   013625
    0006   BLOKF   000001

   EXTERNAL REFERENCES (BLOCK, NAME)

    0007   FACTOR
    0010   DECIDE
    0011   NINTR$
    0012   NRDU$
    0013   NIO2$
    0014   NIO1$
    0015   NEXP1$
    0016   NWDU$
    0017   NSTOP$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   012652 11F      0001   000014 114G     0001   000035 132G     0001   000075 161G
    0001   000156 211G     0001   000172 220G     0001   000176 223G     0001   000210 232G
    0001   000251 247G     0000   012644 5000F    0000   012646 5001F    0000   012650 5002
    0000   012641 5004F    0000   012653 5005F    0000   012677 5006F    0000   012711 5007
    0000   012723 9001F    0003 R 000003 AL       0003 R 000002 AU       0000 R 012621 A1
    0003 R 000004 BU       0000 R 012622 B1       0003 R 000007 CL       0003 R 000006 CU
    0004 R 000000 D        0000 R 012627 DBAR     0003 R 000010 DEL      0000 R 012626 DSUM
    0005 R 000631 ES       0005 R 004563 F        0005 R 000011 FUDGE    0000 R 007020 FX
    0000 R 012632 GB       0000 R 012633 GC       0000 I 012616 I        0003 I 000011 IJ
    0000 I 012637 JFLAG1   0000 I 012630 K        0000 I 012625 L        0000 I 012615 NC
    0003 I 000000 NPAR     0003 I 000001 NPOINT   0000 I 012617 NTIMES   0000 R 003410 RX
    0006 R 000000 SPRED1   0005 R 000000 SSE      0005 R 000000 SX       0000 R 012634 TA
    0000 R 012636 TC       0005 R 004253 X        0005 R 004241 XN


00100      C THIS PROGRAM USES THE ROBERTS AND REED METHOD
00101            DIMENSION X(200),F(9,202),E(9,200),SSE(9),ES(9,200),D(9,3),FUDGE(2
00101           10,20),SE(9),SX(9,200),RX(9,200),FX(9,213)
00103            COMMON/BLOKA/NPAR,NPOINT,AU,AL,BU,BL,CU,CL,DEL,IJ
00104            COMMON/BLOKB/D
00105            COMMON/BLOKC/SSE,FUDGE,ES,XN,SE,X,F,E
00106            COMMON/BLOKF/SPRED1
00106      C     NC=NUMBER OF CYCLES IN FUDGE TABLE
00107            READ(5,5003)NC
00112       5003 FORMAT(I5)
```

```
C       READ IN VALUES OF FUDGE TO BE USED
        DO 2 I=1,NC
        READ(5,5004)FUDGE(5,I),FUDGE(9,I)
 5004 FORMAT(2F10.3)
    2 CONTINUE
        READ(5,9000)NTIMES
 9000 FORMAT(I5)
        DO 9099 IJK=1,NTIMES
        READ(5,5000)A1,B1,C1,AU,AL,BU,BL,CU,CL
 5000 FORMAT(9F5.2)
C       A1,B1,C1,ARE THE INITIAL PARAMETER VALUES FOR THE CENTER
C       AU,AL,BU,BL,CU,CL,ARE THE UPPER AND LOWER BOUNDS RESPECTIVELY ON
C       THE 3 CONTROL PARAMETERS
        READ(5,5001)NPAR,NOBS,SPREDI,DEL
 5001 FORMAT(2I5,2F5.2)
C       NPAR=NUMBER OF PARAMETERS TO BE USED IN THE MODELS
C       NOBS=NUMBER OF TIME PERIODS IN EACH TIME SERIES
C       SPREDI=WIDTH OF THE FACTORIAL ALONG A MAJOR AXIS
        READ(5,5002)(X(I),I=1,NOBS)
 5002 FORMAT(8F10.2)
        NPOINT=((2**NPAR)+1)
C       NPOINT= NUMBER OF DESIGN POINTS
        CALL FACTOR(A1,B1,C1,SPREDI)
C       F(9,200)=FORECASTS AT 9 DESIGN POINTS AND 200 TIME PERIODS
C       E(9,200)=ERRORS AT DESIGN POINTS AND TIME PERIODS
C       SSE(9)=SUM OF SQUARED ERRORS
C       ES(9,200)=ERROR SQUARED AT DESIGN POINTS
        DO 3 I=1,9
        E(I,1)=0.0
        SSE(I)=0.0
        F(I,1)=0.0
        SE(I)=0.0
C       SE=SUM OF ERRORS AT EACH DESIGN POINT
        F(I,12)=X(12)
    3 CONTINUE
        XN=0.0
C       XN=NUMBER OF PERIODS WE HAVE BEEN IN CURRENT PHASE
C       BEGIN THE ACTUAL FORECAST
C       SA(I)=ESTIMATE OF THE LEVEL COMPONENT AT TIME I
C       X(I)=ACTUAL OBSERVATION
C       GA=SMOOTHING CONSTANT,0<GA<1
C       GB=SMOOTHING CONSTANT,0<GB<1
C       GC=SMOOTHING CONSTANT,0<GC<1
C       L=PERIODICITY OF THE SEASON
C       FX(I)=SEASONALITY FACTOR AT TIME I
C       RX(I)=TREND AT TIME I
C       F(K,IJ)=FORECAST FOR TIME IJ
        READ(5,11)L
   11 FORMAT(I5)
        DSUM=0.0
        DO 12 I=1,L
        DSUM=DSUM+X(I)
   12 CONTINUE
        DBAR=DSUM/FLOAT(L)
        DO 13 I=1,L
        DO 14 K=1,NPOINT
   14 FX(K,I+L)=X(I)/DBAR
```

```
00227         13 CONTINUE
00231            DO 15 K=1,NPOINT
00234            SX(K,12)=X(12)/FX(K,24)
00235            RX(K,12)=0.0
00236         15 F(K,13)=(SX(K,12)+RX(K,12))*FX(K,13)
00240            DO 100 IJ=13,40
00243            WRITE(6,5005)
00245       5005 FORMAT(' PERIOD  A1    B1   C1       FORECAST     ACTUAL    DEVIATION
00245          1     ERROR SQ.    SPRED1  SX(K,IJ)  RX(K,IJ)  FX(K,IJ)')
00246            DO 4 K=1,NPOINT
00251            E(K,IJ)=F(K,IJ)-X(IJ)
00252            ES(K,IJ)=(E(K,IJ)**2)
00253            SSE(K)=SSE(K)+(E(K,IJ)**2)
00254            SE(K)=SE(K)+E(K,IJ)
00255            GA=D(K,1)
00256            GB=D(K,2)
00257            GC=D(K,3)
00260            SX(K,IJ)=GA*(X(IJ)/FX(K,IJ))+(1.0-GA)*(SX(K,IJ-1)+RX(K,IJ-1))
00261            RX(K,IJ)=GC*(SX(K,IJ)-SX(K,IJ-1))+(1.0-GC)*RX(K,IJ-1)
00262            FX(K,IJ+L)=GB*(X(IJ)/SX(K,IJ))+(1.0-GB)*FX(K,IJ)
00263            F(K,IJ+1)=(SX(K,IJ)+RX(K,IJ))*FX(K,IJ+1)
00264          4 WRITE(6,5006)IJ,D(K,1),D(K,2),D(K,3),F(K,IJ),X(IJ),E(K,IJ),ES(K,IJ
00264          1),SPRED1,SX(K,IJ),RX(K,IJ),FX(K,IJ)
00303       5006 FORMAT(I4,2X,F6.3,2F5.3,F12.4,F11.4,F12.4,F11.4,F11.4,3F10.4)
00304            WRITE(6,5007)SE(NPOINT),SSE(NPOINT)
00310       5007 FORMAT('SUM OF ERROR= ',F10.2,5X,'SUM OF ERROR SQ. = ',F10.2)
00311            XN=XN+1.0
00312            TA=D(1,1)
00313            TB=D(1,2)
00314            TC=D(1,3)
00315            CALL DECIDE(TA,TB,TC,JFLAG1)
00316            IF(JFLAG1 .EQ. 1)CALL FACTOR(A1,B1,C1,SPRED1)
00320        100 CONTINUE
00322       9999 WRITE(6,9001)
00325       9001 FORMAT(1H1)
00326            STOP
00327            END

      END OF COMPILATION:        NO DIAGNOSTICS.
```

```
WFOR,IS FACTOR
FOR S10A-02/27/73-13:33:24 (,0)

   SUBROUTINE FACTOR    ENTRY POINT 000141

   STORAGE USED: CODE(1) 000161; DATA(0) 000034; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   BLOKA  000012
    0004   BLOKB  000033

   EXTERNAL REFERENCES (BLOCK, NAME)

    0005   NWOU$
    0006   NIO2$
    0007   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000124 1000L     0001    000053 124G     0001    000113 166G     0001    000045 50L
    0003    000003 AL        0003    000002 AU       0003    000005 BL       0003    000004 BU
    0003    000006 CU        0004 R 000000 D         0003    000010 DEL      0003    000011 IJ
    0000 I 000000 K          0003 I 000000 NPAR      0003    000001 NPOINT


00101              SUBROUTINE FACTOR(A,B,C,SPREAD)
00103              DIMENSION D(9,3)
00104              COMMON/BLOKA/NPAR,NPOINT,AU,AL,BU,BL,CU,CL,DEL,IJ
00105              COMMON/BLOKB/D
00105        C     THIS SUBROUTINE FILLS UP THE FACTORIAL DESIGN MATRIX D WITH EACH ROW
00105        C     REPRESENTING A DESIGN POINT. THE CENTER POINT IS ALWAYS IN THE FIRST
00106              D(1,1)=A
00107              D(1,2)=B
00110              D(1,3)=C
00111              IF (NPAR .EQ. 3)GO TO 50
00113          400 D(2,1)=A+SPREAD*(1.0)
00114              D(3,1)=A+SPREAD*(1.0)
00115              D(4,1)=A+SPREAD*(-1.0)
00116              D(5,1)=A+SPREAD*(-1.0)
00117              D(2,2)=B+SPREAD*(1.0)
00120              D(3,2)=B+SPREAD*(-1.0)
00121              D(4,2)=B+SPREAD*(1.0)
00122              D(5,2)=B+SPREAD*(-1.0)
00123              DO 31 K=1,9
00126           31 WRITE(6,98) D(K,1),D(K,2),D(K,3)
00134              GO TO 1000
00135           50 D(2,1)=A+SPREAD*(1.0)
00136              D(3,1)=A+SPREAD*(1.0)
00137              D(4,1)=A+SPREAD*(1.0)
00140              D(5,1)=A+SPREAD*(1.0)
```

```
00141        D(6,1)=A+SPREAD.(-1.0)
00142        D(7,1)=A+SPREAD.(-1.0)
00143        D(8,1)=A+SPREAD.(-1.0)
00144        D(9,1)=A+SPREAD.(-1.0)
00145        D(2,2)=B+SPREAD.(1.0)
00146        D(3,2)=B+SPREAD.(1.0)
00147        D(4,2)=B+SPREAD.(-1.0)
00150        D(5,2)=B+SPREAD.(-1.0)
00151        D(6,2)=B+SPREAD.(1.0)
00152        D(7,2)=B+SPREAD.(1.0)
00153        D(8,2)=B+SPREAD.(-1.0)
00154        D(9,2)=B+SPREAD.(-1.0)
00155        D(2,3)=C+SPREAD.(1.0)
00156        D(3,3)=C+SPREAD.(-1.0)
00157        D(4,3)=C+SPREAD.(1.0)
00160        D(5,3)=C+SPREAD.(-1.0)
00161        D(6,3)=C+SPREAD.(1.0)
00162        D(7,3)=C+SPREAD.(-1.0)
00163        D(8,3)=C+SPREAD.(1.0)
00164        D(9,3)=C+SPREAD.(-1.0)
00165        DO 80 K=1,9
00170     80 WRITE(6,98) D(K,1),D(K,2),D(K,3)
00176     98 FORMAT(//,' MATRIX BEFORE CHECK IS',3F15.3)
00177   1000 RETURN
00200        END

      END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
@FOR,IS DECIDE
FOR S10A-02/27/73-13:33:46 (,0)

   SUBROUTINE DECIDE     ENTRY POINT 000623

   STORAGE USED: CODE(1) 000600; DATA(0) 000073; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    BLOKA   000012
    0004    BLOKB   000033
    0005    BLOKC   013603
    0006    BLOKF   000001

   EXTERNAL REFERENCES (BLOCK, NAME)

    0007    SORT
    0010    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001     000176 11L       0001     000015 110L      0001     000066 113L      0001     000103 114
    0001     000215 12L       0001     000044 124G      0001     000235 13L       0001     000057 133
    0001     000074 144G      0001     000241 20L       0001     000506 2000L     0001     000353 21L
    0001     000412 23L       0001     000431 24L       0001     000451 25L       0001     000500 303
    0003     000003 AL        0003     000002 AU        0000 R 000000 AVGE      0000 R 000026 BIG
    0003     000004 BU        0003     000007 CL        0003     000006 CU        0004 R 000000 D
    0000 R 000011 DIF       0000 R 000027 DMIN      0005 R 010173 E         0000 R 000034 EA
    0000 R 000040 EC        0000 R 000025 EE        0000 R 000037 ELIMN     0000 R 000036 ELI
    0005 R 004563 F         0005 R 000011 FUDGE     0000 I 000023 I         0003 I 000011 IJ
    0000 I 000031 L         0003     000000 NPAR      0003 I 000001 NPOINT    0000 R 000030 RAN
    0005 R 004242 SE        0000 R 000033 SIGMA     0006 R 000000 SPREAD    0005 R 000000 SSE
    0005 R 004253 X         0000 R 000024 XL        0005 R 004241 XN


00101      1*          SUBROUTINE DECIDE(A,B,C,JFLAG)
00103      2*          DIMENSION AVGE(9),SSE(9), FUDGE(20,20),ES(9,200),DIF(9),SE(9),X(20
00103      3*         10),F(9,200),E(9,200),D(9,3)
00104      4*          COMMON/BLOKA/ NPAR,NPOINT,AU,AL,BU,BL,CU,CL,DEL,IJ
00105      5*          COMMON/BLOKB/D
00106      6*          COMMON/BLOKC/SSE,FUDGE,ES,XN,SE,X,F,E
00107      7*          COMMON/BLOKF/SPREAD
00110      8*          IF(XN .GT. 1) GO TO 110
00112      9*          SUMS=0.0
00113     10*          JFLAG=0
00114     11*          GO TO 2000
00115     12*      110 DO 111 I=1,NPOINT
00120     13*      111 AVGE(I)=SSE(I)/XN
00122     14*          XL=XN-1.0
00122     15*    C     XL=NUMBER OF PERIODS SINCE WE CHANGED TO A NEW PHASE
00123     16*          DO 112 I=1,NPOINT
```

```
00126              EE=(SSE(I)-ES(I,IJ))/XL
00127          112 DIF(I)=EE-ES(I,IJ)
00131              BIGD=DIF(1)
00132              DO 113 I=2,NPOINT
00135              IF(DIF(I) .LE. BIGD)GO TO 113
00137              BIGD=DIF(I)
00140          113 CONTINUE
00142              DMIN=DIF(1)
00143              DO 114 I=2,NPOINT
00146              IF(DIF(I) .GE. DMIN) GO TO 114
00150              DMIN=DIF(I)
00151          114 CONTINUE
00153              RANGE=ABS(BIGD-DMIN)
00154              L=XN
00155              S=RANGE*FUDGE(NPOINT,L)
00156              SUMS=SUMS+S
00157              SIGMA=SUMS/XL
00157        C     CALCULATE   EFFECTS
00160              IF(NPOINT .EQ. 9) GO TO 20
00162              EA=0.5*(AVGE(2)+AVGE(3)-AVGE(4)-AVGE(5))
00163              EB=0.5*(AVGE(2)+AVGE(4)-AVGE(3)-AVGE(5))
00164              ELIMP=+3.0*SIGMA/SQRT(XN)
00165              ELIMN=-ELIMP
00166              JFLAG=0
00167              IF(EA .LE. ELIMP) GO TO 11
00171              IF(D(1,1) .LE. 0.1)GO TO 11
00173              A=A-SPREAD
00174              JFLAG=1
00175              GO TO 12
00176           11 IF(EA .GE. ELIMN)GO TO 12
00200              IF(D(1,1) .GE. 0.9)GO TO 12
00202              A=A+SPREAD
00203              JFLAG=1
00204           12 IF(EB .LE. ELIMP)GO TO 13
00206              IF(D(1,2) .LE. 0.1) GO TO 13
00210              B=B-SPREAD
00211              JFLAG=1
00212              GO TO 14
00213           13 IF(EB .GE. ELIMN)GO TO 14
00215              IF(D(1,2) .GE. 0.9) GO TO 14
00217              B=B+SPREAD
00220              JFLAG=1
00221           14 IF(JFLAG .EQ. 1) GO TO 900
00223              GO TO 2000
00224           20 EA=0.25*(AVGE(5)+AVGE(3)+AVGE(2)+AVGE(4)-AVGE(6)-AVGE(7)-AVGE(8)-A
00224             1VGE(9))
00225              EB=0.25*(AVGE(3)+AVGE(2)+AVGE(6)+AVGE(7)-AVGE(5)-AVGE(4)-AVGE(9)-A
00225             1VGE(8))
00226              EC=0.25*(AVGE(6)+AVGE(2)+AVGE(4)+AVGE(8)-AVGE(7)-AVGE(3)-AVGE(5)-A
00226             1VGE(9))
00227              ELIMP=3.0*SIGMA*SQRT(1.0/(2.0*XN))
00230              ELIMN=-ELIMP
00231              IF(EA .LE. ELIMP) GO TO 21
00233              IF(D(1,1) .LE. 0.1) GO TO 21
00235              A=A-SPREAD
00236              JFLAG=1
00237              GO TO 22
```

```
00240       21 IF(EA .GE. ELIMN)GO TO 22
00242          IF (D(1,1) .GE. 0.9) GO TO 22
00244          A=A+SPREAD
00245          JFLAG=1
00246       22 IF(EB .LE. ELIMN)GO TO 23
00250          IF (D(1,2) .LE. 0.1)GO TO 23
00252          B=B-SPREAD
00253          JFLAG=1
00254          GO TO 24
00255       23 IF (EB .GE. ELIMN) GO TO 24
00257          IF(D(1,2) .GE. 0.9)GO TO 24
00261          B=B+SPREAD
00262          JFLAG=1
00263       24 IF(EC .LE. ELIMN)GO TO 25
00265          IF(D(1,3) .LE. 0.1)GO TO 25
00267          C=C-SPREAD
00270          JFLAG=1
00271          GO TO 14
00272       25 IF(EC .GE. ELIMN)GO TO 14
00274          IF(D(1,3) .GE. 0.9) GO TO 14
00276          C=C+SPREAD
00277          JFLAG=1
00300          GO TO 14
00301      900 XN=0.0
00302          DO 901 I=1,NPOINT
00305          SSE(I)=0.0
00306          SE(I)=0.0
00307          F(I,IJ+1)=X(IJ)
00310      901 E(I,IJ)=0.0
00312     2000 RETURN
00313          END

     END OF COMPILATION:        NO DIAGNOSTICS.
```

333445

```
@FOR,IS MAIN
FOR S11A-05/15/73-06:22:25 (,0)

    MAIN PROGRAM

    STORAGE USED: CODE(1) 003601; DATA(0) 011123; BLANK COMMON(2) 000000

    COMMON BLOCKS:

     0003   BLOKA  001602

    EXTERNAL REFERENCES (BLOCK, NAME)

     0004   SIMPLX
     0005   AINTR$
     0006   NRDU$
     0007   NI02$
     0010   NI01$
     0011   NRDU$
     0012   NSTOP$

    STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0000   011020 11F       0001   000023 116G      0001   000042 122G      0001   00013]
     0001   000151 175G      0001   000163 204G      0001   000175 213G      0001   00022:
     0000   011016 5001F     0000   011025 5003F     0000   011025 5005F     0000   01105]
     0000   011067 5006F     0000   011002 7001F     0000   011102 9000F     0000 R 0107/(
     0003 R 001575 AU        0000 R 010777 B         0003 R 001606 BL        0003 R 00157;
     0000 R 011006 DSAF      0000 R 011005 DSUM      0000 R 000310 E         0000 R 01100(
     0000 R 007210 FX        0000 R 011010 GA        0000 R 011011 GB        0000 R 01101:
     0003 I 001601 IJ        0000 I 011013 IL        0000 I 011007 K         0000 I 0107/:
     0003 I 001574 NEW       0000 I 010772 NOBS      0000 I 010771 NPT       0000 I 0107/(
     0000 R 011002 S         0000 R 003650 RX        0000 R 011004 SUM       0000 R 01100:
     0003 R 000014 V         0000 R 000000 X


 00100      1*    C THIS PROGRAM USES THE REGULAR SIMPLEX
 00101      2*          DIMENSION X(200),B(4,3),E(4,220),F(4,220),V(4,220),RX(4,220),SX((
 00101      3*         1220),FX(4,220)
 00103      4*          COMMON/BLOKA/B,V,NEW,AU,AL,BU,BL,IJ
 00103      5*    C NTIMES=NUMBER OF TIME SERIES TO BE ANALYZED.
 00104      6*          READ(5,5007)NTIMES
 00107      7*     5007 FORMAT(I4)
 00110      8*          READ(5,5003)NPT
 00110      9*    C NPT= THE NUMBER OF POINTS USED IN THE SIMPLEX
 00113     10*     5003 FORMAT(I4)
 00114     11*          NOBS=200
 00115     12*          DO 9999 KJ=1,NTIMES
 00120     13*          READ(5,5001) (X(I),I=1,NOBS)
 00120     14*    C X(I)=VALUES OF TIME SERIES
 00126     15*     5001 FORMAT(8F10.2)
 00127     16*          READ(5,11)L
 00132     17*       11 FORMAT(I5)
```

```
00123          READ(5,5000)A,B   ,EDGE,NOBS,AL,AU,BL,BU
00145     5000 FORMAT(2F5.2,F5.3,I5,4F5.2)
00146          D(1,1)=A
00147          D(1,2)=B
00150          P=EDGE*0.9659
00151          G=EDGE*0.2588
00152          D(2,1)=P+A
00153          D(2,2)=G+B
00154          D(3,1)=G+A
00155          D(3,2)=P+B
00155     C D(4,3)=THE DESIGN MATRIX
00156          NEX=0
00157          SUMSQ=0.0
00160          SUM=0.0
00160   C      BEGIN THE ACTUAL FORECAST
00160   C      SX(I)=ESTIMATE OF THE LEVEL COMPONENT AT TIME I
00160   C      X(I)=ACTUAL OBSERVATION
00160   C      GA=SMOOTHING CONSTANT,0<GA<1
00160   C      GB=SMOOTHING CONSTANT,0<GB<1
00160   C      GC=SMOOTHING CONSTANT,0<GC<1
00160   C      L=PERIODOCITY OF THE SEASON
00160   C      FX(I)=SEASONALITY FACTOR AT TIME I
00160   C      RX(I)=TREND AT TIME I
00160   C      F(K,IJ)=FORECAST FOR TIME IJ
00161          DSUM=0.0
00162          DO 12 I=1,L
00165          DSUM=DSUM+X(I)
00166       12 CONTINUE
00170          DBAR=DSUM/FLOAT(L)
00171          DO 13 I=1,L
00174          DO 14 K=1,NPT
00177       14 FX(K,I+L)=X(I)/DBAR
00201       13 CONTINUE
00203          DO 15 K=1,NPT
00206          SX(K,12)=X(12)/FX(K,24)
00207          RX(K,12)=0.0
00210       15 F(K,13)=(SX(K,12)+RX(K,12))*FX(K,13)
00212          DO 100 IJ=13,200
00215          WRITE(6,5005)
00217     5005 FORMAT(' PERIOD   A     B            FORECAST    ACTUAL          DEVIAT
00217         ION      ERROR SQ.   SX(K,IJ)      RX(K,IJ)      FX(K,IJ)')
00220          DO 4 K=1,NPT
00223          E(K,IJ)=F(K,IJ)-X(IJ)
00224          V(K,IJ)=E(K,IJ)**2
00225          GA=D(K,1)
00226          GB=D(K,2)
00227          GC=0.0
00230          RX(K,IJ)=0.0
00231          SX(K,IJ)=GA*(X(IJ)/FX(K,IJ))+(1.0-GA)*(SX(K,IJ-1)+RX(K,IJ-1))
00232          FX(K,IJ+L)=GB*(X(IJ)/SX(K,IJ-1))+(1.0-GB)*FX(K,IJ)
00233          RX(K,IJ)=GC*(SX(K,IJ)-SX(K,IJ-1))+(1.0-GC)*RX(K,IJ-1)
00234          F(K,IJ+1)=(SX(K,IJ)+RX(K,IJ))*FX(K,IJ+1)
00235        4 WRITE(6,5006)IJ,D(K,1),D(K,2),F(K,IJ),X(IJ),E(K,IJ),V(K,IJ),SX(K,
00235         1IJ),RX(K,IJ),FX(K,IJ)
00252     5006 FORMAT(I4,2X,F6.3,1F5.3   ,F12.4,F11.4,E16.5,E16.5,3F10.4)
00253          CALL SIMPLEX(IL,NPT)
00254          WRITE(6,7001)IL
```

```
00257     7001 FORMAT(' THE NEW ROW IS = ',I2)
00260          SUM=SUM+E(IL,IJ)
00261          SUMSQ=SUMSQ+(E(IL,IJ)**2)
00262          WRITE(6,5008)SUM,SUMSQ
00266     5008 FORMAT(' SUM OF ERROR = ',E16.5,5X,'SUM OF ERROR  SQ. = ',E16.5)
00267      100 CONTINUE
00271     9999 WRITE(6,9000)
00274     9000 FORMAT(1H1)
00275          STOP
00276          END

      END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
@FOR,IS SIMPLX
FOR S11A-05/13/73-16:22:29 (,0)
```

```
SUBROUTINE SIMPLX    ENTRY POINT 000272

STORAGE USED: CODE(1) 000307; DATA(0) 001665; BLANK COMMON(2) 000000

COMMON BLOCKS:

0003  BLOKA  001602

EXTERNAL REFERENCES (BLOCK, NAME)

0004   NRDU$
0005   NIO2$
0006   NERR3$

STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001   000026 1L      0001   000110 10L     0001   000237 100L    0001   0000
0001   000076 126G    0001   000124 141G    0001   000131 146G    0001   0001
0001   000251 204G    0001   000220 35L     0001   001225 36L     0000   0015
0003 R 001576 AL      0003 R 001575 AU      0000 R 001607 BIG     0003 R 0016
0003 R 000000 D       0000 R 001613 ONE+    0000 R 001612 G       0000 I 0015
0000   001636 INJP$   0000 I 001611 J       0000 I 001605 KNTR    0000 I 0016
0000 I 001604 NP      0000 R 001572 PE      0000 R 000000 S       0003 R 0000
```

```
00101          SUBROUTINE SIMPLX(IL,NPT)
00103          DIMENSION D(4,3)          ,S(10),V(9,220),XV(9,220),RE(10)
00104          COMMON/BLOKA/D,V,NEW,AU,AL,BU,BL,IJ
00105          NP=NPT-1
00106          KNTR=0
00107          DO 5 I=1,NPT
00112        5 XV(I,IJ)=V(I,IJ)
00114        1 BIG=XV(1,IJ)
00115          KNTR=KNTR+1
00116          IF(KNTR .GT. (NPT+1)) WRITE(6,6002)
00121          IF(KNTR .GT. 4)GO TO 111
00123     6002 FORMAT(1H0,'HELP')
00124          N=1
00125          DO 10 I=2,NPT
00130          IF(BIG .GE. XV(I,IJ))GO TO 10
00132          BIG=XV(I,IJ)
00133          N=I
00134       10 CONTINUE
00136          IF(N .EQ. NEW) GO TO 35
00140          DO 20 I=1,NP
00143       20 S(I)=0.0
00145          DO 21 I=1,NP
00150          DO 21 J=1,NPT
```

BIBLIOGRAPHY

1. Box, G. E. P., "Evolutionary Operation: A Method for Increasing Industrial Productivity," Applied Statistics, Vol. 6, No. 2 (June, 1957), pp. 81-101.

2. Box, G. E. P., and Draper, N. R., Evolutionary Operation, John Wiley & Sons, Inc., New York, 1969.

3. Chow, W. M., "Adaptive Control of the Exponential Smoothing Constant," Journal of Industrial Engineering, Vol. 16, No. 5 (Sept. - Oct. 1965).

4. Montgomery, D. C., "Adaptive Control of Exponential Smoothing Parameters by Evolutionary Operation," AIIE Transactions, Vol. 11, No. 3 (September, 1970).

5. Nelder, J. A., and Mead, R., "A Simplex Method for Function Minimization," Computer Journal, Vol. 7, No. 4 (January, 1965), pp. 308-13.

6. Roberts, S. D., and Reed, R., Jr., "The Development of a Self-Adaptive Forecasting Technique," AIIE Transactions, Vol. 1, No. 4, December, 1969.

7. Roos, C. F., "Survey of Economic Forecasting Techniques," Econometrica, October, 1955, pp. 363-95.

8. Spendley, W., Hext, G. R., and Humsworth, F. R., "Sequential Application of Simplex Designs in Optimization and Evolutionary Operation," Technometrics, Vol. 4, No. 4 (November, 1962).

9. Trigg, D. N., and Leach, A. G., "Exponential Smoothing with an Adaptive Response Rate," Operational Research Quarterly, Volume 18, No. 1, (March, 1967), pp. 53-9.

10. Winters, P. R., "Forecasting Sales by Exponentially Weighted Moving Averages," Management Science (April, 1960), pp. 324-342.